

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**

**DIPLOMOVÁ PRÁCE**

**2009**

**Oldřich Vašut**

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Systém pro cílený e-mail marketing**  
**Direct mail system**

**2009**

**Oldřich Vašut**

# Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.  
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Čeladné dne 7.5.2009

Oldřich Vašut

# Poděkování

Rád bych na tomto místě poděkoval vedoucímu diplomové práce Ing. Davidu Ježkovi, Ph.D. za odbornou pomoc. Nemalý dík patří Martinu Bednářovi za odborné rady a podnětné připomínky při realizaci tohoto projektu.

# Abstrakt a klíčová slova

## Abstrakt:

Tato diplomová práce se zabývá návrhem a implementací systému pro cílený e-mail marketing. Systém umožňuje vytváření a správu marketingových kampaní, jejich vyhodnocování a také samotné rozesílání reklamních emailů. Systém je řešen pomocí moderních nástrojů a aplikačních rámců. Práce rovněž seznamuje čtenáře s problematikou e-mail marketingu a poskytuje přehled technologií, použitých při vývoji projektu. Patří mezi ně následující: platforma Eclipse RCP pro vývoj desktopových aplikací, nástroj Maven pro správu a automatické sestavování Java projektů, JavaMail API pro práci s elektronickou poštou, nástroj Hibernate zajišťující objektově relační mapování, webový aplikační rámec Wicket a další.

## Klíčová slova:

e-mail marketing, kampaň, e-mail, adresář, odběratel, Java, Eclipse RCP, Maven, Hibernate, JavaMail, Wicket, Hessian, systém, klient, server

# Abstract and keywords

## Abstract:

This diploma thesis deals with design and implementation of the direct mail system. System provides creating and managing marketing campaigns as well as their evaluation and distribution of the promotional e-mails. System is solved by using modern tools and frameworks. This material also acquaints reader with issues of e-mail marketing and provides an overview of technologies used for development. This overview includes technologies such as following: Eclipse RCP platform for developing desktop applications, Maven tool for Java project management and build automation, JavaMail API used to work with electronic mail, object relation mapping tool Hibernate, web framework Wicket and others.

## Keywords:

e-mail marketing, campaign, e-mail, mailing list, recipient, Java, Eclipse RCP, Maven, Hibernate, JavaMail, Wicket, Hessian, system, client, server

# Seznam použitých symbolů a zkratek

AJAX – Asynchronous JavaScript And XML  
API – Application Programming Interface  
AWT – Abstract Window Toolkit  
CAN-SPAM – Controlling the Assault of Non-Solicited Pornography And Marketing Act of 2003  
CRUD – Create, read, update and delete operace  
CSS – Cascading Style Sheets  
DAO – Data Access Object  
DMA – Direct Marketing Association  
GEF – Graphics Ediding framework  
HQL – Hibernate Query Language  
HTML – HyperText Markup Language  
HTTP – HyperText Transfer Protocol  
IDE – Integrated Development Environment  
IMAP – Internet Message Access Protocol  
JDBC – Java Database Connectivity  
JDT – Java Development Tools  
JAF – Java Activation Framework  
JAR – Java ARchive  
Java EE – Java Enterprise Edition  
JFace – sada nástrojů uživatelského rozhraní na platformě Elipse  
JSF – Java Server Faces  
kB – Kilobyte, jednotka informace  
MIME – Multipurpose Internet Mail Extensions  
MVC – Model View Controller  
OPT-IN – opt in e-mail, přihlášení k příjmu emailové inzerce  
OPT-OUT – opt out e-mail, odhlášení z příjmu emailové inzerce  
ORM – Object-Relational Mapping  
OS – Operating system  
OSGi - Open Services Gateway initiative  
PDA – Personal Digital Assistant  
PDE – Plug-in Development Environment  
POJO – Plain Old Java Object  
POM – Project Object Model  
POP3 – Post Office Protocol version 3  
RAP – Rich AJAX Platform  
RCP – Rich Client Platform  
SMTP – Simple Mail Transfer Protocol  
SOAP – Simple Object Access Protocol  
SQL – Structured Query Language  
SWT – Standard Widget Toolkit  
UI – User Interface

URL – Uniform Resource Locator

USA – The United States of America

WYSIWYG – What You See Is What You Get

XHTML – eXtensible Hypertext Markup Language

XML – eXtensible Markup Language

# Obsah

<b>1 Úvod .....</b>	<b>9</b>
<b>2 E-mail marketing .....</b>	<b>10</b>
<b>2.1 Charakteristika e-mail marketingu .....</b>	<b>10</b>
2.1.1 Typy e-mail marketingu .....	11
<b>2.2 Využití e-mail marketingu .....</b>	<b>11</b>
<b>2.3 Výhody a nevýhody .....</b>	<b>12</b>
2.3.1 Výhody .....	12
2.3.2 Nevýhody .....	12
<b>2.4 Emailové taktiky a strategie .....</b>	<b>13</b>
2.3.1 Použitelnost e-mail marketingu .....	13
2.3.2 Návrhy pro psaní působivých nabídek .....	14
2.3.3 Příprava kampaně a její vylepšení .....	14
2.3.4 Emailové kampaně přizpůsobené pro mobilní zařízení .....	15
<b>2.4 Vzhled a návrh e-mailů .....</b>	<b>16</b>
<b>2.5 E-mail marketingové statistiky .....</b>	<b>16</b>
<b>2.6 Správa seznamu odběratelů .....</b>	<b>17</b>
<b>2.7 Poskytovatelé e-mail marketingových služeb .....</b>	<b>18</b>
<b>2.8 Zákony upravující nevyžádanou reklamu .....</b>	<b>18</b>
2.8.1 CAN-SPAM .....	18
2.8.2 Přihlášení k emailové inzerci <i>OPT-IN</i> .....	18
2.8.3 Odhlášení z emailové inzerce <i>OPT-OUT</i> .....	19
2.8.4 „Antispamový“ zákon .....	19
<b>3 Použité technologie .....</b>	<b>20</b>
<b>3.1 Eclipse RCP .....</b>	<b>21</b>
3.1.1 Úvod .....	21
3.1.2 Vlastnosti .....	22
3.1.3 Architektura .....	23
3.1.4 SWT a JFace .....	25
3.1.5 UI Workbench .....	26
3.1.6 Zhodnocení .....	26
<b>3.2 Maven .....</b>	<b>27</b>
3.2.1 Co je Maven? .....	27
3.2.2 Životní cyklus .....	28
3.2.3 POM .....	28
<b>3.3 Java Mail .....</b>	<b>31</b>
3.3.1 Architektura .....	31

<b>3.4 Hibernate</b>	<b>32</b>
3.4.1 Co je Hibernate?	32
3.4.2 Architektura	32
3.4.1 Výhody a nevýhody	32
<b>3.5 Wicket</b>	<b>33</b>
3.5.1 Co je Wicket?	33
3.5.2 Vlastnosti	33
3.5.3 Zhodnocení	34
<b>3.6 Hessian a Burlap</b>	<b>34</b>
<b>4 Řešení systému pro cílený e-mail marketing</b>	<b>37</b>
<b>4.1 Úvod</b>	<b>37</b>
<b>4.2 Analýza a návrh</b>	<b>37</b>
4.2.1 Požadavky na systém	37
4.2.2 Funkční celky	38
4.2.3 Serverový modul	39
4.2.4 RCP klient	41
4.2.5 Servletový modul	45
4.2.6 Webové statistiky	45
<b>4.3 Životní cyklus kampaně</b>	<b>46</b>
<b>4.4 Nasazení</b>	<b>48</b>
4.4.1 Model nasazení systému	48
4.4.2 Instalace a konfigurace	49
<b>4.5 Implementační detaily</b>	<b>49</b>
<b>5 Závěr</b>	<b>51</b>



# 1 Úvod

Cílem této práce je navrhnout a implementovat systém pro cílený e-mail marketing. E-mail marketing se stal v současnosti mocným nástrojem mnoha obchodníků. Jelikož v dnešním světě informačních technologií a progresivního obchodu má své významné místo reklama, jsou nástroje zajišťující její běh klíčové. Inzerce a reklama na nejrůznější výrobky a služby nás obklopuje na každém kroku a ovlivňuje podvědomě tak, až si to mnohdy ani neuvědomujeme. Co e-mail marketing dnes znamená a co všechno bude výsledný systém umožňovat je popsáno v následujících kapitolách. Ve zkratce, tento systém bude poskytovat rozesílání komerčních či nekomerčních emailových zpráv na specifikovaný seznam odběratelů. Rozesílané zprávy budou součástí emailové kampaně. Systém pak umožní komplexní správu těchto kampaní společně s jejich statistickým vyhodnocováním.

V první části práce je osvětlen pojem e-mail marketing a jeho význam ve světě obchodu. Jsou zde rozebírány otázky proč jej využít, jaké přináší výhody a nevýhody a jaké jsou emailové taktiky a strategie. Součástí této kapitoly je i nastínění právních zákonů CAN-SPAM a Zákonu o některých službách informační společnosti, známý jako „antispamový“ zákon, starající se o nevyžádanou reklamu.

Druhá část práce nastiňuje technologie použité při vývoji systému. Jedná se zejména o aplikační rámec, používaný k vývoji „tlustých“ klientů Eclipse RCP, v němž je realizována klientská část aplikace. V této části jsou představeny i zbylé aplikační rámce a nástroje použité při vývoji systému jako jsou JavaMail, Hibernate, Wicket či Maven. Je zde vysvětleno, proč byly použity a jaké jsou jejich přednosti.

Další kapitolou, plynule navazující na prezentované technologie, je řešení samotného systému. Řešení obsahuje podkapitoly týkající se analýzy a návrhu systému, princip na jakém funguje, životní cyklus kampaně a je zde popsána celková architektura systému s jednotlivými funkčními moduly.

V závěru práce je pak shrnuto celé řešení s jeho zhodnocením. Jsou zde nastíněny vyhlídky na další vývoj projektu, jak by se mohl v budoucnu rozšiřovat a zlepšovat.

## 2 E-mail marketing

### 2.1 Charakteristika e-mail marketingu

V dnešním světě informačních technologií a progresivního obchodu má své významné místo reklama. Inzerce a reklama na nejrůznější výrobky a služby nás obklopuje na každém kroku a ovlivňuje podvědomě tak, až si to mnohdy ani neuvědomujeme. Co je tedy e-mail marketing a jak souvisí s reklamou? *E-mail marketing* je formou přímého „direct“ marketingu. Obecnější přímý marketing je pod-disciplínou marketingu, který se vyznačuje rozesíláním zpráv přímo zákazníkům, aniž by k tomu potřeboval zprostředkující média. Mezi přímý marketing patří následující tři služby: *direct mail* neboli reklamní pošta, doručovaná klasickou poštou do domu, *telemarketing* známý též jako dotazování po telefonu a *e-mail marketing*. E-mail marketing využívá k této výměně zpráv mezi obchodníkem a zákazníkem emailovou komunikaci. V širším slova smyslu může být každý e-mail, zaslaný potencionálnímu či současnému zákazníkovi, považován za e-mail marketing. Použití e-mailu k propagování nabízeného zboží či služeb snadno podpoří náš obchod a navíc prostřednictvím internetu a e-mailu lze dosáhnout reklamy a prosadit svou nabídku na mezinárodní úrovni. Hlavní komponentou email marketingu je emailová marketingová kampaň, která ke svému běhu potřebuje několik podpůrných zdrojů a to především samotnou nabídku (např. ve formě HTML dokumentu), seznam odběratelů, časově vymezené období platnosti kampaně, nezbytné softwarové a hardwarové vybavení a samozřejmě strategický plán. E-maily mohou být navrženy takovým způsobem, aby pomohly zákazníkovi upevnit důvěru a oddanost, například rozšířením zkušenosti s obchodníkem pomocí zvláštních nabídek, kupónů či pozorností podniku.

E-mail marketing je velice podobný přímé reklamě, katalogům, tištěným věstníkům a jiným reklamním materiálům rozesílaných klasickou poštou. Elektronická forma marketingu má oproti té klasické několik výhod, zejména jde o získání zpětné vazby zákazníka, více o výhodách v kap. 2.3.1. Nicméně lidé, čelící nevyžádané poště v klasické i elektronické formě, mají na email marketing negativní pohled. Byla by proto velká škoda a nepřesnost vidět za pojmem e-mail marketing nevyžádanou poštu známou jako „spam“. V e-mail marketingu se nikdy nerozesílají e-maily s nabídkami zákazníkům, kteří o takový typ informací nežádali. V opačném případě je na takové nabídky nahlíženo jako na „spam“ a společnost se může ocitnout na seznamu adres rozesílajících „spam“, případně jí mohou být zablokovány účty a webové stránky a jméno takové společnosti ztratí na důvěryhodnosti, nehledě nato, že v mnoha zemích je tato činnost protizákonní. Za tímto účelem je třeba mít vždy jistotu, že máme od zákazníků či potencionálních zákazníků svolení k zasílání marketingových e-mailů. Zajištěno to může být například za políčkem na webové stránce pro zatrnutí volby, že si zákazník přeje dostávat v budoucnosti tyto informační materiály. Více o nevyžádané poště a zákonech toto upravujících v kap. 2.8.

Email marketing se v posledních letech stal velmi populární metodou komunikace. Je značně levnější než ostatní druhy komunikace a je zaměřen přímo na zákazníka. Pokud je udělán důkladně, je velmi úspěšnou metodou v marketingu ve světě obchodu. Problematika e-mail marketingu je blíže rozebírána v [1],[2] a [9].

### 2.1.1 Typy e-mail marketingu

**Přímý e-mail (*direct e-mail*)** – tento typ e-mailu obsahuje propagační zprávu, jenž má za úkol přesvědčit současného či potencionálního zákazníka, aby zavítal na naše webové stránky a daný produkt si koupil. Takové sdělení může obsahovat speciální nabídku, kupón či jakékoliv jiné lákadlo pro zákazníka. Tyto zprávy mohou být zasílány pouze zákazníkům, jejichž kontakty jsme legálně získali z našich webových stránek či od partnerské společnosti. Pak mohou být zacíleny přímo na zákazníka.

**Zadržovací e-mail (*Retention e-mail*)** – tyto e-maily směřují k existujícím zákazníkům za účelem jejich udržení. Mohou být formou bulletinu, nabídek či jiných propagačních metod. Tento typ e-mailu by měl zákazníkovi podávat cenné informace zábavnou formou, nikoliv pouze zprávy o prodeji.

**Inzerce v e-mailech ostatním lidem (*Advertising In Others People's Email*)** – v této metodě inzerent platí jiným firmám nebo spolupracuje s partnerskou společností a umisťuje své reklamy v jejich e-mailech, posílaných jejich odběratelům.

## 2.2 Využití e-mail marketingu

Jak bylo naznačeno v předešlé kapitole, mnoho obchodníků nedůvěřuje použití e-mail marketingu kvůli zmatku, týkající se legálnosti a nevyžádané pošty. Lidé pak žijí s mylným dojmem, že e-mail marketing, propagující obchod, není nic víc, než „spamování“, což ovšem není pravda. Společnosti využívají služeb e-mail marketingu, rozesílají nabídky lidem, kteří jim k tomu dali patřičný souhlas. Pokud společnosti následují toto pravidlo, pak se e-mail marketing stává mocným nástrojem. Jeho podstata na fungování by se dala shrnout do několika bodů. Je zaměřen přímo na zákazníky, vše je řízeno daty, pomáhá spravovat prodej a lze vybudovat oddanost a důvěru ve vztahu mezi společností a zákazníkem. Prodej navíc může být podporován i jinými metodami.

Návrhem emailových kampaní s pomocí integrace a segmentace databází zákazníků, či jiných technik, lze zlepšit způsob, jak e-maily zacílit. Mohou být rozeslány v závislosti na geografické poloze, pohlaví, věku, či jiných pokročilých možnostech a lze tak za běhu generovat emaily přizpůsobené konkrétním zákazníkům nebo skupině s danými vlastnostmi. Emaily tedy mohou být přizpůsobeny parametry, které si sami vymezíme. S každou odeslanou kampaní jsme schopni sbírat potřebná data k analýze, aby další odeslaná nabídka byla přesnější a efektivnější. Čím více jsou emailové kampaně kultivovanější, tím úspěšnější reakci můžeme očekávat od zákazníků. Hlavními cíli e-mail marketingu jsou: zvýšení prodeje, generování průzkumů a zvýšení návštěvnosti webových stránek. Směřováním zákazníka na webové stránky jsme mu schopni nabídnout daný produkt či službu. To lze podpořit podrobnějšími informacemi či přihlášením na propagační akce, čím více e-mail informuje, tím více se lidé budou zajímat o nabídku obchodu. Pokud přiložíme novinky a nabídky jiných partnerských společností, vybudujeme si podvědomí, povzbudíme důvěru a posílíme vztahy se všemi zákazníky. Zůstává zde jedna citlivá záležitost při rozesílání e-mailů. Musíme si být vždy jisti, že seznam příjemců je ten, na který chceme dané informace odeslat. Pokud se někdo odhlásí z odběru e-mailů, musíme respektovat jeho přání a odebrat jej ze seznamu. Taktiky a strategie pro emailové kampaně lze vždy zlepšit a těžit tak z úspěchů konzistentní práce.

## 2.3 Výhody a nevýhody

### 2.3.1 Výhody

E-mail marketing získal u firem na oblibě především díky následujícím výhodám. Emailový seznam odběratelů poskytuje distribuci informací širokému spektru specifických a potencionálních zákazníků za relativně nízké náklady. V porovnání s jinými náklady na média, jako tištěná reklama je e-mail méně nákladný. Přesná návratnost investic může být sledována a může prokázat, že je vysoká, když je udělána správně. E-mail marketing je často označován za druhou nejefektivnější marketingovou online taktiku po *Search marketingu* (což je zviditelňování webových stránek vyhledávacím strojem a umísťování reklamy na ně odkazující). Doručení emailové zprávy zabere kratší dobu (sekundy či minuty) v porovnání s tištěnou reklamou (jeden či více dnů). Inzerent je schopen propagovat zprávu mezi jeho příznivce, kdežto reklama, založená na webu závisí na tom, zda zákazník stránky navštíví nebo ne. Emailové zprávy jsou snadno sledovatelné. Inzerent může sledovat uživatele přes *autoresponders* (programy zajišťující automatickou odpověď na e-mail), *web bugs* (objekt na stránce či v e-mailu, sloužící k zjištění, zda daný uživatel viděl stránku nebo e-mail), *bounce messages* (nedoručená zpráva), *unsubscribe requests* (požadavek na odhlášení ze seznamu), *read receipts* (zpráva o doručení e-mailu), *click-through* (metoda na měření úspěšnosti online kampaně pomocí poměrů kliků na odkazy) a další. Tyto mechanismy mohou být použity k měření *open rates* (poměrová metoda použita k zjištění, zda uživatel viděl, či otevřel e-mail), kladných či záporných odpovědí a ke korelaci prodeje s marketingem. Inzerent může generovat opakovatelně své obchodní nabídky pohodlně a automaticky. Tím může dosáhnout značného počtu odběratelů, kteří dali svolení k této komunikaci. Z předchozích průzkumů vyplývá, že více než polovina uživatelů internetu si během běžného dne pošle či zkontroluje e-mail. V e-mail marketingu mohou být využity specifické typy interakcí, kterou jsou navázány na odeslané zprávy a mohou automaticky posílat další zprávy, nebo upravovat profil odběratele. Dalším bodem, který lze zařadit mezi výhody, je že e-mail marketing je ekologický a šetří spotřebu papíru oproti tištěné reklamě.

### 2.3.2 Nevýhody

Mezi nevýhody e-mail marketingu se dá zařadit pouze jedna a to, že kromě společností využívající jej k legální komunikaci se zákazníci, existují i společnosti a jednotlivci, rozesílající nevyžádanou poštu, známou jako „spam“.

Pro běžného uživatele je poměrně obtížné odlišit „spam“ od legálního email marketingu. Je to zapříčiněno zejména následujícími důvody. Společnosti rozesílající „spam“ se představují jako legitimní provozovatelé. Dále pak zpočátku stále se měnící zákony, jako například největší politické sdružení, starající se o přímý marketing v USA, *Direct Marketing Association (DMA)* přinutila zákonodárce zlegalizovat některé aktivity, které někteří poskytovatelé internetu považují za „spamování“, jako je rozesílání nevyžádané pošty, která sice obsahuje „opt-out“ mechanismus. V naší zemi je situace trochu odlišná, více o zákonech upravující nevyžádanou poštu v kapitole 2.8. Objem „spamu“ ve světě zapříčiňuje, že někteří uživatelé pak považují i legální e-mail marketing za „spam“. Kvůli velkému objemu „spamu“ na internetu jsou spamové filtry nezbytné pro většinu uživatelů. Někteří obchodníci pak hlásí, že jejich legální komerční emailové zprávy jsou často zachyceny

skrytými filtry. Na druhou stranu je téměř výjimkou, že by si běžný uživatel stěžoval na spamové filtry, které zachycují i legální komerční zprávy.

Společnosti zvažující využití email marketingu se musí ujistit, že jejich program neodporuje zákonům, upravujícím jeho použití v dané zemi.

Ve spojených státech toto zajišťuje zákon nazývaný jako *CAN-SPAM (United States Controlling the Assault of Non-Solicited Pornography and Marketing Act)*, v Evropské unii pak *European Privacy and Electronic Communications Regulations 2003*, v České republice je to *Zákon o Některých službách informační společnosti*, případně i jiné ustanovení a politiky poskytovatelů internetu. I přesto, že inzerent dodržuje zákony, může se vyskytnout na listině společností, rozesílající nevyžádanou poštu. Pokud ji tak administrátoři označí, dá se tomu ovšem bránit.

## 2.4 Emailové taktiky a strategie

Mnoho společností používá stále propracovanější strategie, udržující jejich produkty a služby aktuální a zajímavé. Jejich cílem je využít e-mail k udržení pozornosti zákazníka.

Pomocí e-mailu lze dosáhnout více zaměřením se na různé části databáze zákazníků a oslovit tak lidi, kteří o nás ví či neví. Analýzou zákazníků s jejich nakupovacími zvyky lze rozdělit seznam na části a pak podle analýzy postupovat v dalších nabídkách. Příkladem rozdělení může být, jak drahé zboží zákazníci kupují či jiné specifické vlastnosti. Čas od času je třeba přezkoumat a analyzovat části na web stránkách, obsahující přihlášení k odběru propagačních e-mailů a způsob odhlašování, pokud potřebují změny.

Klíčová je také údržba seznamu odběratelů. Některé taktiky počítají s nečinností uživatelů oproti pravidelným zákazníkům, vytvoří tak zajímavější e-maily a omezí jejich smazání jako nepřečetné. Rozdělení seznamu na části je zapotřebí testovat a kontrolovat, zda je informace, zasláná zákazníkovi, pro něj zajímavá. Úspěšnost lze změřit počtem získaných odezev. Rozdíly je možné vyzkoušet na dvou částech seznamu, kde na každý rozešleme nepatrně změněnou zprávu a pak sledovat, která lépe funguje.

Ukončené emailové kampaně nejsou příliš dlouho efektivní. Analyzují se za běhu či hned po ukončení kampaně a na základě pozitivních reakcí můžeme vyrukovat s novou a zajímavější, abychom udrželi zájem zákazníka.

Osobní kontakt poskytují e-maily, jejichž zaslání je spuštěno nějakou událostí. Příkladem může být přání k narozeninám zákazníka, tím se bude cítit výjimečně a posílí se vztah mezi ním a prodejcem. Záleží jen na tom, jak využijeme informace z databáze zákazníků, vše samozřejmě v mezích zákona.

### 2.3.1 Použitelnost e-mail marketingu

Ke zvážení použitelnosti systému pro e-mail marketing je velmi důležitá správa báze zákazníků. Ovlivňují ji tři následující faktory: Zachování jednoduchého přihlašovacího procesu, kde jsou webové stránky tomuto procesu přizpůsobeny a měli bychom mít připraven plán pro přihlašování zákazníků.

Dalším faktorem je přehledný obsah našeho e-mailu, se všemi nezbytnými informacemi, přitom přitažlivý a dostatečně jednoduchý. Třetím faktorem je povolení změn zákazníkovi bez nepříjemností. Zákazník nemá být na pochybách, zda jsme mu poskytli dostatek informací při přihlášení, možnost odhlášení ze seznamu a mít možnost se seznámit s naší politikou a se vším co si myslíme, že má vědět.

### 2.3.2 Návrhy pro psaní působivých nabídek

Když se spustí emailová kampaň, očekáváme, že si odběratelé najdou čas a otevrou si e-mail. To je ovšem zatím jen polovina úspěchu. Pouze polovina z nich pak reaguje na nabídku. Mnoho tedy záleží na její podání, tak aby upoutala co nejvíce zákazníků. Nabídka obsahuje pět klíčových komponent, ze kterých se skládá.

- **Viditelnost** – výzva k akci (např. výzva k navštívení webových stránek) musí být dobře viditelná a neměla by splývat s textem, může být i na více místech v e-mailu.
- **Srozumitelnost** – jako u všech nabídek i u emailové formy platí, že zákazník má mít vše jasné, aby nebyl odrazen.
- **Přinucení** – přinutit uživatele, aby chtěl vědět více a následoval odkazy, např. poskytnutí výhod po přejití na webové stránky.
- **Naléhavost** – nenechávat nabídku neomezenou. Uživatelům dáme časově omezený úsek k reagování. Lidé snadněji reagují, pokud jsou řízeni termíny.
- **Opakování** – zopakováním nabídky na stránce, na kterou uživatel přešel z e-mailu, pomáháme uživateli si zapamatovat proč přesně klikl na daný odkaz.

### 2.3.3 Příprava kampaně a její vylepšení

Po přípravě nabídky a vyčlenění seznamu, na který ji chceme rozeslat, je třeba před spuštěním kampaně zvážit několik následujících bodů, aby firma neztratila na serióznosti.

- **Ověření seznamu e-mailů** – ověříme, že zpráva bude posílána správným lidem, kteří nám k tomu navíc dali oprávnění.
- **Bližší náhled na hlavičku** – jde zejména o pole v e-mailu „Od“, „Předmět“ a dále musíme vybrat vhodnou dobu pro rozeslání.
- **Pozorná kontrola těla e-mailu** – měli bychom si být jisti, že e-mail obsahuje odkaz na jeho zobrazení online, titulek či nadpis je umístěn nápadně, personalizace e-mailu je korektní, počáteční odstavec je pro čtenáře zajímavý, měl by mít srozumitelný obsah i bez načtení obrázků, jednoduchý a snadný k přečtení a ne příliš dlouhý.

- **Snadno použitelný návrh** – je vhodné se ujistit, že nepoužíváme příliš mnoho grafických objektů, kvůli pomalému načítání, poskytnuté odkazy jsou skutečně funkční, zkontrolovat správnost zobrazení v různých emailových klientech, případně zda máme připravenou verzi s čistým textem namísto HTML dokumentu
- **Kontrola patičky** – na závěr zkontrolujeme funkčnost odkazu pro odhlášení, připojení fyzické adresy, kontaktních údajů, odkaz na webovou stránku či poznámku *copyright*.

Tři následující body se zaměřují na vylepšení emailové kampaně, tak aby byla pro zákazníka atraktivní a neodrazovala jej.

Seznam e-mailů je dobré rozdělit spíše podle chování zákazníků než demograficky. Kampaně mohou být založeny na analýzách předchozích a můžeme sledovat návyky kupujících namísto demografického rozdělení (např. pohlaví, věk, atd.) anebo lze tyto informace kombinovat. Sledováním vlastností jako, které emaily byly otevřeny a přečteny, které odkazy byly nejnavštěvovanější, budeme mít jasný obraz o dalších kampaních založených na aktuálních informacích o zákazníkovi.

Prozkoumání, co lidi vyhledávají na internetu a analyzování informací, které by mohly být užitečné pro marketingovou kampaň. I při emailových kampaních nelze zapomenout, jak lidé našli náš web, jestli nás našli přes reklamní bannery, pomocí vyhledávacího stroje, přes jiné webové stránky, v komentářích blogů či pomocí jiných aktivit vyhledávacích strojů. Tyto aktivity lze poté analyzovat a případně kombinovat s e-mail marketingem.

Sledováním a analyzováním odezvy na emailové marketingové kampaně lze poté vyhodnotit, která je lepší. Zavedením různých metrik prověříme, co přináší nové zákazníky a co je naopak odvádí pryč. Kontrolu můžeme zaměřit například na odkazy zajímaví zákazníkům nejvíce.

### 2.3.4 Emailové kampaně přizpůsobené pro mobilní zařízení

V dnešní době, kdy jsou mobilní zařízení typu PDA a mobilní telefony s přístupem k internetu de facto standard, měli bychom zvážit přizpůsobení emailové kampaně těmto zařízením. Stejně jako webové stránky, optimalizované pro mobilní zařízení, řídí se i emailové kampaně stejnými doporučeními. V mobilních zařízeních jsme omezeni velikostí displeje, které zobrazují stránky odlišně a rychlostí připojení k internetu, email tedy musí být menší velikosti, aby se co nejvíce zkrátila doba načítání. Pro plnou kompatibilitu mezi zařízeními existuje i možnost připravit email jak ve formě HTML dokumentu, tak i textové podobě. Kteří zákazníci využívají ke čtení pošty mobilní zařízení, lze například zjistit poskytnutím této volby při přihlášení do seznamu odběratelů. Při optimalizaci je třeba dodržet následující:

- přeformátování textu, tak aby byl čitelný a přehledný i na malém displeji (doporučuje se 20-40 znaků na řádek a 12-15 řádků na obrazovku)
- zjednodušení URL. Mobilní zařízení zobrazují adresní řádek 4-5 řádků na obrazovku, pak je třeba použít textovou verzi. Musíme si být ale vědomi, že zkrácením URL odkazů můžeme přijít o některé metody sledování.

- zvážení přizpůsobení i webových stránek pro mobilní zařízení, tak aby zákazník mohl přejít na stránky a přitom se mu zobrazily správně.

## 2.4 Vzhled a návrh e-mailů

E-maily navržené pro kampaň by měly být zejména snadné ke čtení, jednoduché k procházení, s nenáročnou grafikou a přitom profesionálně vyhlížející. Aby byly e-maily pro zákazníka přitažlivé, platí pravidlo 1-5-10. Zákazníkovi stačí jedna sekunda, aby viděl o čem e-mail je, během pěti sekund je třeba upoutat jeho pozornost a za deset sekund procházení e-mailem zjistí, zda je to, co jsme mu sdělili, pro něj zajímavé. Při návrhu emailových zpráv postupujeme trochu odlišným způsobem, než při návrhu webových stránek. Důležité je si uvědomit několik technických a informačních prvků k přípravě dobrého HTML e-mailu.

- neplýtvat čtenářovým časem – použít vše-vypovídajícího předmětu e-mailu, případně použít shrnutí na začátku zprávy
- ujistit se, že máme oprávnění poslat e-mail zákazníkovi, více v kap. 2.8.2
- nikdy neposílat informace, které nejsou relevantní nebo které čtenář neočekává
- poskytnout uživateli odkaz pro odhlášení ze seznamu odběratelů, více v kap. 2.8.3
- obrázky v emailu nepoužívat na významných místech, protože někteří uživatelé mohou mít zakázáno zobrazování obrázků nebo jej musí dodatečně povolit
- v některých případech je výhodnější použít tabulek k vystavění designu HTML e-mailu, protože někteří klienti neumí zpracovávat plovoucí prvky
- použití vložených stylů namísto externího CSS souboru, protože někteří klienti mohou ignorovat stylování bloků v HTML dokumentu, na něž je použito CSS stylování
- připravení verze e-mailu s čistým textem, protože ne všichni klienti jsou schopni zobrazit HTML dokument
- emaily nesmí porušovat zákony dané země, proto před jeho odesláním musíme překontrolovat zda splňuje „opt-in“ přihlášení, zda obsahuje legální jméno a emailovou adresu v poli „Od“, přesný a výstižné pole „Předmět“, validní adresu pro odpověď, která může sloužit jako odhlašovací mechanismus.

## 2.5 E-mail marketingové statistiky

Za několik let využívání e-mail marketingu bylo odborníky zjištěno následující. Mnoho lidí reaguje příznivěji na email s kratším předmětem (méně než 50 znaků) a se snadno čitelným obsahem více doplňujícími odkazy. Poměr kliků (*click-through*, tj. poměr mezi kliky na odkazy v e-mailu) těchto zpráv je vyšší než těch co mají méně odkazů. Velikost zprávy pak nemá vliv na poměr kliků. Emaily s kratším předmětem od 0 do 49 znaků mají větší poměr otevření o 12,5% a o 75% větší poměr kliků než zprávy s předmětem 50 znaků s více. Rozdíly mezi poměry otevření a kliky nejsou tak velké. Emaily obsahující 25 a více odkazů byly otevřeny o 12% více než ty, které mají méně než 25. Poměr kliků byl o 29% vyšší pro e-maily s více než 25 odkazy. Skutečná velikost e-mailu nehraje příliš velkou roli, spíše to, co obsahuje a jak vypadá. Například zprávy menší než 3kB mají největší poměr otevření, odmítnutí a odhlášení. Průměrná velikost e-mailu se pohybuje mezi 20-79kB s poměry otevření a kliky kolem 3,9%. Optimální velikost zprávy by měla být kolem 40-50kB pro snadné přijetí, což se v budoucnosti bude narůstat se zrychlujícím se připojením k internetu. Zkoumání zda



HTML či e-mail s čistým textem je výhodnější, vychází podle statistik nastejno. HTML e-maily jsou však vizuálně přitažlivější a v dnešní době nejrozšířenější. Verzi e-mailu s čistým textem mnoho společností přikládá k HTML e-mailu jako alternativu. Příkladem může být průzkum, který prokázal, že odeslané HTML emaily o 55% větší poměr kliků než e-maily s čistým textem. Oproti tomu e-maily odeslané s velmi bohatým HTML byly o 39% méně úspěšné než emaily s čistým textem.

## **2.6 Správa seznamu odběratelů**

Získat seznam kontaktů s emailovými adresami lze několika způsoby, vždy však musíme mít jistotu, že postupujeme v mezích zákona a máme nezbytné právní podklady. Nejrozumnější je si vybudovat seznam vlastními prostředky, např. pomocí registračního formuláře na našich webových stránkách, který obsahuje souhlas s přidáním do seznamu. Dalším způsobem, jak lze seznam získat, je si jej koupit od jiné společnosti. I v tomto případě musí obsahovat dostatečná povolení k použití seznamu. Samozřejmě vše musí být právně podloženo, abychom se vyhnuli nákupu adres z pochybného zdroje, které mohou být posbírány nelegálně posbírány z webu.

### **Strategie pro neaktivní odběratele**

Existuje několik návrhů, jak obnovit zájem neaktivních odběratelů, přestože to není snadné. Pomoci může rozdělení seznamu do dvou skupin - aktivních a neaktivních uživatelů a pro každou skupinu navrhnout jiný obsah. Druhým způsobem je rozeslání průzkumu, např. o společnosti, co je třeba zlepšit a jaké zprávy by si přáli dostávat. V neposlední řadě bychom si měli ověřit, zda je adresa platná a uživatel ji stále používá, případně se pokusit o její aktualizaci.

### **Efektivní proces odhlašování**

Schopností odhlásit se z emailové kampaně je klíč k úspěchu. Dáváme tím najevo, že přečtení emailu je na dobrovolné bázi. Poskytneme mu tedy snadnou možnost odhlášení, např. pomocí připojeného odkazu. Navíc v mnoha zemích je toto uzákoněno.

### **Segmentování seznamu pomocí sledování kliků**

Segmentace seznamu odběratelů podle různých hledisek je jedním ze způsobů, jak zefektivnit použití emailových marketingových kampaní. Rozdělení pak může být demografické či založené na chování nebo kombinace obojího. Společnosti používají vesměs jednoduchou segmentaci, protože složitější zabere více času a navíc mnoho společností ani neví jak ji provést. Segmentaci seznamu je nutné si předem rozmyslet, může být jednoduchá nebo komplexní v závislosti na druhu informace, známé ze seznamu. Jestliže máme segmentaci seznamu vytvořenou, můžeme připravit zprávy dynamicky pro každý segment, ty potom nevypadají tolik mechanicky připravené. Tyto segmenty se dají vybudovat na základě analýzy sledování kliků.

Pozornost je potřeba věnovat také zprávám, které nedorazily a byly odrazeny zpět. V tomto případě je nutné zkontrolovat překlepy, validitu adres, prohlašování odběratelů, spamové filtry a jiné.

## 2.7 Poskytovatelé e-mail marketingových služeb

Všechny předem zmiňované věci si mnoho odborníků uvědomuje a především tu, že e-mail marketing je výnosný obchod. V současnosti proto existuje spousta společností poskytující komplexní e-mail marketingové služby. Společnosti se liší spektrem a kvalitou nabízených služeb a cenovou nabídkou. Většina těchto poskytovatelů nabízí poradenský servis a webové rozhraní pro správu kampaní a seznamu odběratelů.

## 2.8 Zákony upravující nevyžádanou reklamu

Jak již bylo v předešlém textu zmíněno, je rozesílání obchodních sdělení upraveno patřičnými zákony. Ve spojených státech, kde se e-mail marketing primárně rozšířil, se o nevyžádanou reklamu stará zákon *CAN-SPAM* z roku 2003. U nás, s obdobou platí od roku 2004 Zákon o některých službách informační společnosti, který vyhovuje směrnici Evropské unie o soukromí a elektronické komunikaci. Zákony se během času mění, proto je nutné sledovat jejich změny a přizpůsobovat se jim.

### 2.8.1 CAN-SPAM

*CAN-SPAM* neboli plným názvem *Controlling the Assault of Non-Solicited Pornography And Marketing Act of 2003* upravuje mimo jiné také rozesílání komerční a nevyžádané pošty. Služby, které splňují podmínky, kladené zákonem, musí při rozesílání e-mailů dodržovat následující: počáteční ověření uživatele s platnou emailovou adresou, poskytnutí odhlášení a je zakázán nákup adres s nedostatečným oprávněním k použití. K naplnění podmínek zákony dodržují společnosti následujících tří pravidel:

**Odhlášení** – zajištění viditelného a funkčního mechanismu pro odhlášení ve všech e-mailech, následně odběratelův požadavek na odhlášení musí být vyřízen do 10ti dnů, seznam odhlášených uživatelů může být poté použit, jen pokud to neodporuje zákonu.

**Obsah** – přesná specifikace polí „Od“, „Předmět“, musí být uvedena legální adresa inzerenta či publikujícího a zpráva musí být opatřena označením, pokud je určena pouze pro dospělé.

**Způsob rozesílání** – zprávy nesmí být rozesílány přes *open relay* (SMTP server nastavený tak, že kdokoli jej může využít k rozesílání pošty), nesmí být odeslány na nelegálně získané adresy, zpráva nesmí obsahovat falešnou hlavičku, na druhou stranu ze zákony jsou vyjmuty zprávy náboženské, politické a zprávy splňující marketingové mechanismy v specifikované v zákoně či zprávy národní obrany.

Ve spojení s tímto zákonem je potřeba uvést dva hlavní mechanismy zajišťující přihlašování (*subscribe*, *OPT-IN*) a odhlašování odběratelů (*unsubscribe*, *OPT-OUT*).

### 2.8.2 Přihlášení k emailové inzerci *OPT-IN*

Tato metoda je založena na poskytnutí souhlasu uživatele k přijímání inzerce a k zařazení do seznamu odběratelů. Metoda eliminuje nevýhody e-mail marketingu a může mít několik forem.

**Nepotvrzovaný OPT-IN** – nový odběratel poskytne svou emailovou adresu (např. přes web) bez jakéhokoli dalšího ověření, zda mu uvedená adresa patří. Toto způsobuje, že poslaný e-mail může být považován za „spam“, jelikož může přitít na adresu náležící jiné osobě. Toho se totiž dá zneužít jinými nekalými společnostmi k podvrhování emailových adres na seznam odběratelů.

**Potvrzovaný OPT-IN** – novému uživateli je po požádání zaslána potvrzovací zpráva, že se jedná o danou osobu. Tímto se sice vyhneme situaci s podvrhovanými adresami, ale na druhou stranu tato metoda přináší jisté nepohodlí uživatele, spojené s vyřízením potvrzovací zprávy. Metoda je známá také jako potvrzení přihlášení, uzavřená smyčka OPT-IN či jako dvojitý OPT-IN.

### **2.8.3 Odhlášení z emailové inzerce *OPT-OUT***

Tato metoda je opakem metody pro přihlášení OPT-IN a je používána k odebrání emailové adresy ze seznamu odběratelů. Inzerent musí zajistit mechanismus pro odhlašování z odběru inzerce. V praxi pak každá zpráva obsahuje přiložený odkaz, pomocí kterého se může uživatel kdykoliv odhlásit

### **2.8.4 „Antispamový“ zákon**

V českém právním řádu je řešena otázka boje proti nevyžádaným obchodním sdělením zákonem č. 480/2004 Sb. O některých službách informační společnosti, který vychází z direktiv Evropské unie, konkrétně ze směrnic 200/31/ES o elektronickém obchodu a 2002/58/ES o soukromí v elektronických komunikacích.

Tento zákon je velice podobný zákonu *CAN-SPAM*, lišící se některými pojmy, ale se stejným principem. Nejdůležitějším pojmem je „prokazatelný souhlas“, jenž musíme od uživatele obdržet. Důležitá není ani tak technická forma, jako spíše prokazatelnost, že byl souhlas projevem svobodné vůle.

I když se zákony v daných zemích neustále upravují a přijímají nové, lze vyhovění zákonům shrnout do dvou bodů. Obchodník dodržuje metodu OPT-IN a vlastní souhlasy od uživatelů pro přijímání inzerce. Dále pak dodržuje i druhou metodu OPT-OUT, tj. poskytnutí možnosti uživateli se kdykoliv z odběru odhlásit. Po dodržení všech ustanovených podmínek se e-mail marketing stává silným obchodním nástrojem.

## 3 Použité technologie

Následující kapitola poskytuje náhled na technologie, použité při vývoji projektu. Každá z následujících podkapitol by jistě vydala svým obsahem na nejednu samostatnou publikaci, avšak pro účely této práce jsou jednotlivé technologie pouze představeny a zběžně popsány. Navrhovaný systém pro správu e-mail marketingových kampaní se dotýká několika technických a informačních oblastí. Pro jeho komplexnější charakter bylo k implementaci použito několika různých technologií a aplikačních rámců. Využití frameworků třetích stran s sebou přináší několik výhod, především snadnější implementaci, s tím související menší časovou náročnost a zpřehlednění architektury. Tyto výhody lehce vyvažuje nutná znalost těchto technologií a čas strávený jejich studiem. O výběru nástrojů, nabízených třetími stranami, rozhoduje neméně jejich cenová dostupnost. Z tohoto důvodu jsou pro vývoj projektu vybrány pouze volně dostupné nástroje, jejichž použití je upraveno *open source* licencí. Konkrétnější zdůvodnění jednotlivých nástrojů je popisováno v dalších kapitolách. Použitelnost a složitost daných nástrojů hrály také významnou roli při výběru. Všechny vybrané nástroje spojuje společný programovací jazyk Java. Jak je jistě známo, Java je platformě nezávislá a může tak pracovat na mnoha operačních systémech. Tento fakt je z hlediska budoucí použitelnosti serverové i klientské části velmi významný.

Následující seznam shrnuje některé použité knihovny, aplikační rámce a technologie:

- **Eclipse RCP** – platforma pro vývoj „tlustých“ klientů, využita při tvorbě Administrační konzole
- **JavaMail** – standardní rozhraní pro práci s elektronickou poštou, využití pro modul rozesílání emailů jednotlivým odběratelům
- **Hibernate** – ORM nástroj, využitý pro mapování doménového modelu systému do databáze
- **Wicket** – Java webový framework, použitý pro vývoj statistické části systému
- **Maven** – nástroj pro podporu a sestavování projektu, použitý k sestavování všech projektů
- **Trac** – je rozšířený „wiki“ nástroj, spojený se systémem pro sledování problémů projektu
- **Hessian** – binární protokol webových služeb, který má spíš blíže k vzdálenému volání metod, použitý ke komunikaci mezi klientem administrační konzole a serverem
- **OSCache** – systémová cache, použita ke cachování dotazů do databáze
- **Springframework** – komplexní aplikační rámec, využita pouze jeho část pro modul servletů

## 3.1 Eclipse RCP

### 3.1.1 Úvod

Vývojové prostředí Eclipse s sebou přináší podporu pro vytváření vizuálně bohatých desktopových aplikací ve formě *open-source* projektu Eclipse RCP (*Rich Client Platform*). Volbu, který nástroj využít k vývoji klientské části systému administrační konzole, pro správu e-mail marketingových kampaní, vyhrál právě Eclipse RCP. Samotný Eclipse v dnešní době není pouze vývojovým prostředím, ale především platformou. Účelem Eclipse platformy je poskytnout služby, nezbytné pro integraci softwarových vývojových nástrojů, které jsou implementovány formou plug-inů. Návrh Eclipse lze chápat jako framework pro plug-iny, tedy kromě poměrně malého runtime jádra je vše ostatní plug-in nebo množina souvisejících plug-inů. V tomto návrhu spočívá síla snadné rozšiřitelnosti Eclipsu. Eclipse RCP nabízí alternativní možnost ke knihovnám, jako je např. AWT či Swing, vytvořit aplikace tlustých klientů v Javě. Je součástí open-source projektu Eclipse Project. Eclipse od svého počátku prošel řadou změn, inovací a v současné době je v procesu dynamického vývoje. Informace týkající se vývoje na platformě Eclipse RCP jsem čerpal z [3] a [4].

### Historie

Eclipse Project má za sebou relativně bohatou historii a komunita jeho příznivců a přispěvatelů stále roste. Eclipse spadá pod sdružení Eclipse Foundation, nad kterou drží záštitu několik firem v čele s IBM spolu s HP, Intel, MontaVista, QNX Software Systems, SAP, Serena Systems a další.

- Eclipse 1.0 (2001) – první vydání vývojového prostředí (IDE), objevily se stížnosti na funkcionalitu, absence některých funkcí
- Eclipse 2.1 (2003) – vylepšení vývojového prostředí, vzniká komunita vývojářů aplikací založených na Eclipse IDE
- Eclipse 3.0 (2004) – vznik Eclipse RCP, z původního Eclipse byly odstraněny závislosti na IDE a dalších komponentách jako jej Help a Update, vznik RCP API pro podporu konfigurace a životního cyklu aplikace
- Eclipse 3.1 (2005) – vylepšení nástrojů pro vývoj, nové a rozšířené API

Od roku 2006 Eclipse Foundation publikuje každý rok nové vydání, které obsahuje Eclipse Platform a velký počet dalších Eclipse projektů.

- Eclipse 3.2 (2006) - Calisto
- Eclipse 3.3 (2007) - Europa
- Eclipse 3.4 (2008) - Ganymede
- Eclipse 3.5 (2009) – Galileo

### 3.1.2 Vlastnosti

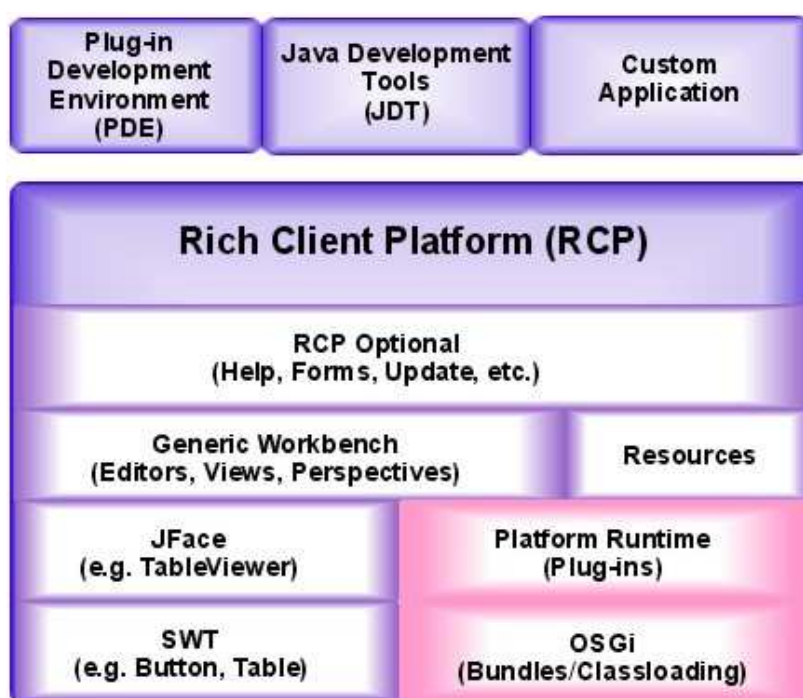
Proč je tedy Eclipse RCP s oblibou využíván k tvorbě desktopových aplikací? Jak tomu napovídá předešlá kapitola, jeho výhoda spočívá v samotné architektuře. Následující souhrn vlastností zároveň ukazuje pozitivní aspekty využití:

- **Komponenty** - Eclipse zahrnuje robustní komponentní model. Systémy, založené na Eclipse jsou vystavěny vytvářením komponent, zvaných plug-iny. Plug-iny jsou verzovány a mohou být sdíleny více aplikacemi. Společně může být nainstalováno více verzí stejného plug-inu a aplikace může být nakonfigurována na verzi, kterou potřebuje. Tento přístup je atraktivní, jelikož umožňuje rozvíjení aplikací přidáváním a nahrazováním komponent.
- **Střední vrstva a infrastruktura** – Na vrcholu komponentního modelu stojí řada frameworků, které usnadňují vývoj klientských aplikací. Eclipse RCP je v podstatě střední (*middleware*) vrstvou v architektuře klientské aplikace. Zahrnuje mechanismy jako paradigma UI (*User Interface* - uživatelské rozhraní), škálovatelnost UI, rozšiřitelnost aplikace, podporu nápovědy, síťovou aktualizaci, zpracování chyb a mnoho dalšího. Navíc nás tyto mechanismy nutí dodržovat vrstvenou architekturu a správné programátorské techniky při vývoji aplikace.
- **Přirozená uživatelská zkušenost** – Mnoho uživatelů vyžaduje vizuální bohatost klientu, pohodlné ovládání a jsou neochotní měnit své přirozené návyky. Jemné, vstřícné a ovladatelné uživatelské rozhraní poskytuje Eclipse SWT (*Standard Widget Toolkit*) a JFace sadou grafických nástrojů na platformě Java, více o SWT a JFace v kapitole 3.1.4.
- **Sofistikovaná instalace a aktualizace** – framework komponent umožňuje plug-inům nasazení a aktualizaci pomocí několika mechanismů: http, Java Web Start, Update Sites, jednoduché kopírování souborů anebo sofistikované „enterprise“ systémy pro správu.
- **Obsluha při odpojení** – Jelikož klientské aplikace jsou spuštěny na místních počítačích, mohou pracovat bez síťového připojení. To je hlavní výhoda oproti tenkým klientům, spouštěných přes webové rozhraní. Aplikace mohou být odpojeny od sítě a pracovat s místními replikami databází a obsah se může aktualizovat až po obnovení připojení.
- **Podpora vývojových nástrojů** – Eclipse IDE obsahuje podporu pro vývoj RCP aplikací, což zahrnuje nástroje pro vývoj, testování a jejich balení.
- **Knihovny komponent** – Framework komponent obsahuje plug-iny pro sestavení rozšiřitelných UI, správu obsahu nápovědy, podporu instalace a aktualizace, editaci textu, konzoly, komponenty pro grafickou editaci (*GEF Graphics Editing framework*), manipulaci s daty a další.

### 3.1.3 Architektura

#### Model běhové podpory

Platforma Eclipse je tvořena jádrem zajišťující běhovou podporu *Runtime*, pracovním prostředím *Workbench*, pracovním prostorem *Workspace*, nápovědou *Help* a komponentami pro týmovou spolupráci *Team Support*. Hlavním úkolem runtime je nalézt dostupné plug-iny v Eclipse adresáři. Každý plug-in má XML manifest soubor, ve kterém jsou definovány plug-iny vyžadované aplikací. Ty v sebe zahrnují body rozšíření *extension points* pro ostatní plug-iny a rozšiřující body z ostatních plug-inů. Počet plug-inů může být velký a proto nejsou jádrem načítány, dokud nejsou požadovány, čím se minimalizuje čas startu aplikace a sníží se nároky na zdroje aplikace.



Obr. 3.1 Architektura Eclipse RCP

#### Workspace

Workspace je odpovědný za správu uživatelských prostředků, které jsou organizovány do jednoho nebo více projektů na nejvyšší úrovni. Každý projekt koresponduje s podadresářem v adresáři workspace, může obsahovat soubory a složky. Normálně je každá složka podadresářem v adresáři projektu, ale může být i odkazem mimo workspace do adresáře souborového systému. Workspace udržuje nízko-úrovňovou historii změn každého projektu. To umožňuje okamžité navrácení změn nebo navrácení se k poslední uložené stavu. Návrat ke změnám může být i několik dní starý, což záleží na uživatelském nastavení. Historie pak minimalizuje riziko ztráty uložených prostředků. Workspace je také zodpovědný za oznámení zainteresovaným nástrojům o změnách zdrojů.

## Workbench

Workbench je grafické uživatelské prostředí. Stará se o vykreslování známých menu, nástrojových lišt, o organizaci perspektiv, které obsahují jednotlivé pohledy a editory více v kapitole 3.1.5.

## Team Support

Team Support je plug-in, usnadňující použití verzovacího a konfiguračního systému ke správě zdrojů v uživatelských projektech. Definuje workflow, nezbytné pro ukládání a vybírání zdrojů z úložiště.

## Help

Eclipse v sebe zahrnuje rozšiřitelnou komponentu pro podporu nápovědy. Vývojáři aplikací mohou tvořit dokumentaci v HTML formátu s použitím navigace, definované pomocí XML. Připojováním jednotlivých plug-inů k ostatním lze vkládat nová témata nápovědy k již existujícím.

## Organizace plug-inů

Jak bylo popsáno v předchozích kapitolách, Eclipse hraje v zásadě roli komponentního frameworku. Pro vývoj aplikace v Eclipse RCP je dobré si zhruba uvědomit architekturu množiny plug-inů jej tvořící. Na obrázku 3.1 tvoří části SWT a JFace, OSGi a Runtime sady plug-inů, které formují základ RCP. Plug-iny jsou mezi sebou vzájemně provázány.

Správa závislostí pak tvoří podstatnou část práce při sestavování aplikace. Plug-iny popisují samy sebe a obsahují seznam plug-inů a funkcí, které potřebují ke své práci. Úkolem runtime je pak řešit tyto závislosti a sjednocovat plug-iny dohromady. Zajímavé je, že závislosti mezi plug-iny nejsou kvůli samotnému Eclipse, ale vycházejí implicitně z kódu a struktury plug-inů. Eclipse pomáhá vytvářet explicitní závislosti a efektivně je spravovat.

## Druhy aplikací

Pomocí Eclipse můžeme de facto vytvořit tři typy aplikací:

- SWT aplikace – je podobná JFC/Swing aplikacím, nemůžeme použít Eclipse RCP Workbench
- Eclipse plug-in – může plně využívat RCP Workbench a může být nasazena v RCP aplikaci
- Eclipse RCP – balíček mnoha plug-inů, je nasazena jako samostatná aplikace

## OSGi a Equinox framework

Komponentní model plug-inů v Eclipse je založen na implementaci specifikace OSGi frameworku. OSGi specifikace vytváří framework pro definování, skládání a spuštění komponent či balíku (*bundle*). Bundle lze považovat za implementaci plug-inů a není mezi těmito elementy žádný zásadní či funkcionální rozdíl. Obojí jsou mechanismy pro sjednocování a správu obsahu. OSGi frameworky pak spravují balíky a jejich kód tak, že získávají potřebný classloader, pro načtení dané třídy. Classpath každého plug-inu je dynamicky konstruována podle závislostí uvedených v manifest souboru (`MANIFEST.MF`). V kontextu Eclipse je role OSGi frameworku spojit dohromady



instalované plug-iny a povolit jim interakci a spolupráci. Pečlivá správa závislostí a classpaths zajistí řízení komunikace mezi plug-iny a vytváření systému je flexibilní a snadno rozšiřitelné.

Vlastnosti OSGi kvůli kterým je integrován do Eclipse:

- pojmenování a seskupování částí do balíčků
- správa závislostí
- rozhraní pro importy, exporty a updaty
- vestavěná bezpečnostní vrstva
- dynamičnost

S příchodem RCP v Eclipse 3.0 v sebe zahrnoval Equinox Technology Project. Equinox se stal certifikovanou implementací OSGi specifikace a umožňuje přizpůsobit runtime dynamické podpoře instalace a odinstalace plug-inů bez nutnosti restartovat aplikaci.

### **Extesion / Extension points**

OSGi specifikace poskytuje mechanismus pro definování a spouštění oddělených komponent. Vrstva runtime k tomu přidává mechanismus pro definování vazeb mezi plug-iny tzv. *extension registry*. Plug-iny tak mohou být rozšiřitelné pomocí definovaných bodů rozšíření *extension points*, které jsou popsány v XML formátu. Buďto můžeme rozšířit funkcionality RCP komponenty pomocí extension pointu, nebo je můžeme využít k rozšíření funkcionality naší komponenty.

### **3.1.4 SWT a JFace**

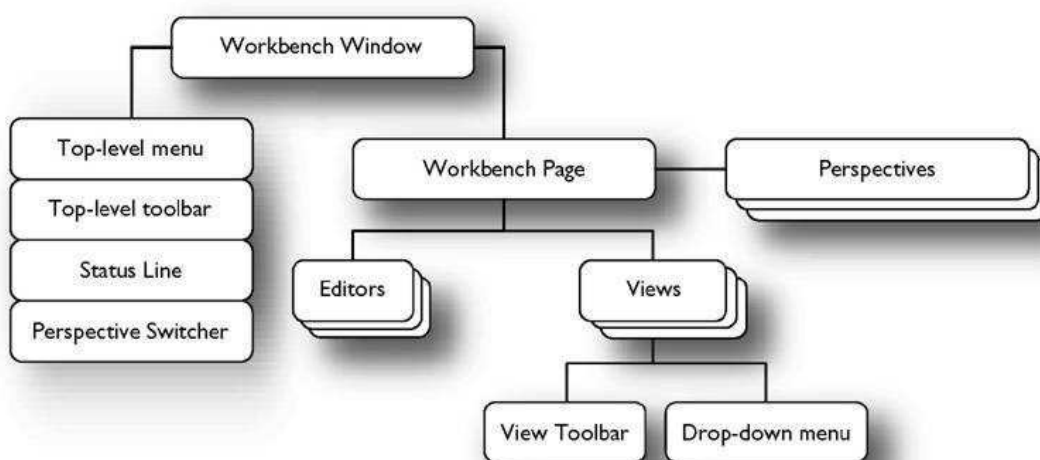
Mezi hlavní výhodu RCP jako platformy pro tvorbu tlustých klientů patří velké množství již hotových grafických komponent. Eclipse používá dvou knihoven s grafickými komponentami SWT (Standard Widget Toolkit) a JFace.

SWT knihovna zajišťuje nezávislost komponenty na platformě a jejich nativní vzhled. Na rozdíl od standardních grafických Java API, jako jsou AWT a Swing, emulující nativní grafické nástroje, SWT je mapováno na grafické knihovny operačního systému. Znamená to, že SWT musí být portováno na každou platformu, kterou Eclipse podporuje. Prakticky to již takový problém není, protože existují portace do platform Windows, Linux/Motif, Linux/GTK2, Solaris, QNX, AIX, HP-UX a Mac OS X. SWT je nízko-úrovňová grafická knihovna, poskytující standardní ovládací prvky uživatelského rozhraní, jako seznamy, menu, fonty, barvy, tlačítka, popisky a mnoho dalšího. Jelikož je SWT propojena přímo na grafické knihovny OS, má aplikace přirozený vzhled, ať je spuštěna na kterékoliv platformě.

JFace je grafický framework, nezávislý na systémových knihovnách a tvoří tak další vrstvu nad SWT. Poskytuje vysoko-úrovňové komponenty, jaké jsou např. textová podpora, dialogy, průvodci, menu, nástrojové lišty, stavový řádek a mnoho dalších. Tvoří tak nadstavbu na SWT, která zjednodušuje práci s nízko-úrovňovými. Všechny komponenty striktně dodržují návrhový vzor MVC (*Model View Controller*). Eclipse nás ovšem nenutí používat SWT či JFace, můžeme klidně pracovat s AWT či Swing.

### 3.1.5 UI Workbench

Stejně tak jako JFace přináší strukturu k SWT, Workbench přidává prezentaci a koordinaci mezi prvky do JFace. Workbench poskytuje uživatelské rozhraní s rozmístěnými pohledy a editory a ostatními grafickými prvky. Workbench poskytuje deklarativní rozšíření elementů uživatelského rozhraní. To znamená, že některé grafické komponenty lze připojit k naší aplikaci deklarativním způsobem, pomocí bodů rozšíření. Ty zaobalují prvky JFace a mají základní dopad na rychlost a škálovatelnost uživatelského rozhraní. Jelikož jsou všechny závislosti vyhodnocovány líným principem (komponenty jsou načteny až když se opravdu potřebují), zvýší se rychlost. Bez deklarativního rozšíření s rostoucím počtem UI komponent narůstá velikost kódu a čas potřebný k načítání.



Obr. 3.2 Rozvržení Pohledů, Editorů a Perspektiv v Runtime, zdroj: [3]

V našem plug-inu může existovat pouze jedna instance Workbench. Ta může obsahovat několik oken Windows. Každé okno obsahuje komponentu page. Workbench povoluje uspořádat grafické položky do složek, které lze vyskládat na pracovní plochu. Každá objekt page může obsahovat více perspektiv, které slouží jako vizuální kontejner pro pohledy views a editory editors. Perspektiva umožňuje jednotlivé uživatelské úlohy pomocí omezené množiny pohledů, editorů, akcí, odkazů na průvodce, související pohledy a perspektivy.

### 3.1.6 Zhodnocení

Eclipse RCP není pouze jediná existující platforma pro vytváření aplikací tlustých klientů v Javě. Mezi další RCP platformy patří např. NetBeans a Spring RCP. Všechny tyto platformy poskytují robustní základ pro tvorbu RCP aplikací a jsou si velmi podobné. Následující seznam ukazuje faktory ovlivňující výběr RCP platformy:

- podpora cílového OS pro provoz výsledné aplikace
- předchozí znalost prostředí
- dokumentace a velikost komunity

Mezi výhody Eclipse RCP patří:

- profesionální funkcionalita bez nutnosti instalace – vývojář má k dispozici mnoho již hotových komponent, jako systém nápovědy, aktualizace, modul pro nastavení aplikace a další
- architektura Eclipse RCP – připojování nových funkcí do aplikace a jejich načítání bez nutnosti restartu aplikace
- profesionální uživatelské rozhraní – profesionální souprava grafických komponent
- rychlý běh aplikace – grafická knihovna SWT je úzce spjata s daným OS
- dynamický vývoj Eclipse Projektů – lze očekávat i v budoucnu podporu a nové komponenty

Eclipse RCP má i své nevýhody, kterých ovšem není zdaleka takový počet jako výhod:

- velikost výsledné aplikace – přidáváním vyžadovaných plug-inů narůstá velikost aplikace
- časová náročnost – čas nutný k seznámení s technologií

## 3.2 Maven

### 3.2.1 Co je Maven?

Při vývoji podnikových *Enterprise* Java aplikací je proces sestavování, balení a nasazení více komponent v různých prostředích náročný. Tyto komponenty mohou záviset na dalších komponentách třetích stran. Navíc statické zdroje, konfigurační soubory a popisy pro nasazení musí být spravovány tak, aby mohl být vydán produkt do cílového prostředí. Maven je nástroj, který toto vše obstará. Slouží k správě projektů a jejich automatickému sestavování. Funkcionalitou je podobný nástroji Ant, ovšem má daleko jednodušší konfigurační model a nabízí širší možnosti. Maven tak poskytuje možnosti, jak pomoci vyvíjet znovupoužitelné a účinné přístupy k sestavování softwaru. Více informací o produktu Maven lze nalézt na jeho oficiálních stránkách [10]. Hlavními cíli Maven, pomáhající vývojáři zpřehlednit správu projektu v krátké časové době, jsou:

- vytvoření snadnějšího procesu sestavení – Maven eliminuje nutnou znalost základních mechanismů a odstiňuje nás od detailů.
- poskytnutí jednotného sestavovacího systému – Projekt je sestavován pomocí POM (*Project Object Model*) a množiny plug-inů, které jsou sdíleny Mavenem přes všechny projekty.
- poskytnutí projektových informací – Maven umí generovat mnoho užitečných informací o projektu, které pocházejí z nadefinovaného POM a ze zdrojů projektu. Maven poskytuje např. log změn v projektu, křížené odkazování zdrojů, adresář, seznamy závislostí, zprávy z jednotkových testů i s pokrytím a další.
- poskytnutí návrhu pro dobré návyky při vývoji – Cílem Maven je také shromáždit současné principy, dobré návyky při vývoji. Specifikace, vykonávání a reportování výsledku jednotkových testů patří k normálnímu cyklu Maven. Dalšími cíli je správa verzí jednotlivých vydání projektu, sledování problémů a ukazuje, jak by měla vypadat struktura projektu.

- povoluje transparentní migraci nových prvků – Maven poskytuje jednoduchý způsob aktualizace své instalace díky komponentní architektuře formou plug-inů.

Jestliže máme nutnost v našem projektu spravovat závislosti na knihovnách třetích stran, pak toto je jedna ze silných vlastností Mavenu. Jeho jádro umí dynamicky stáhnout knihovny z repository. Maven vlastní centrální repository, která poskytuje přístup k mnoha verzím různých open source Java projektů. Jakmile jsou knihovny staženy do lokální repository na disku, není nutné je při další práci znova stahovat. Maven neposkytuje podporu jen pro získávání souborů z repository ale lze nahrávat nové verze projektu po sestavení do repository.

### 3.2.2 Životní cyklus

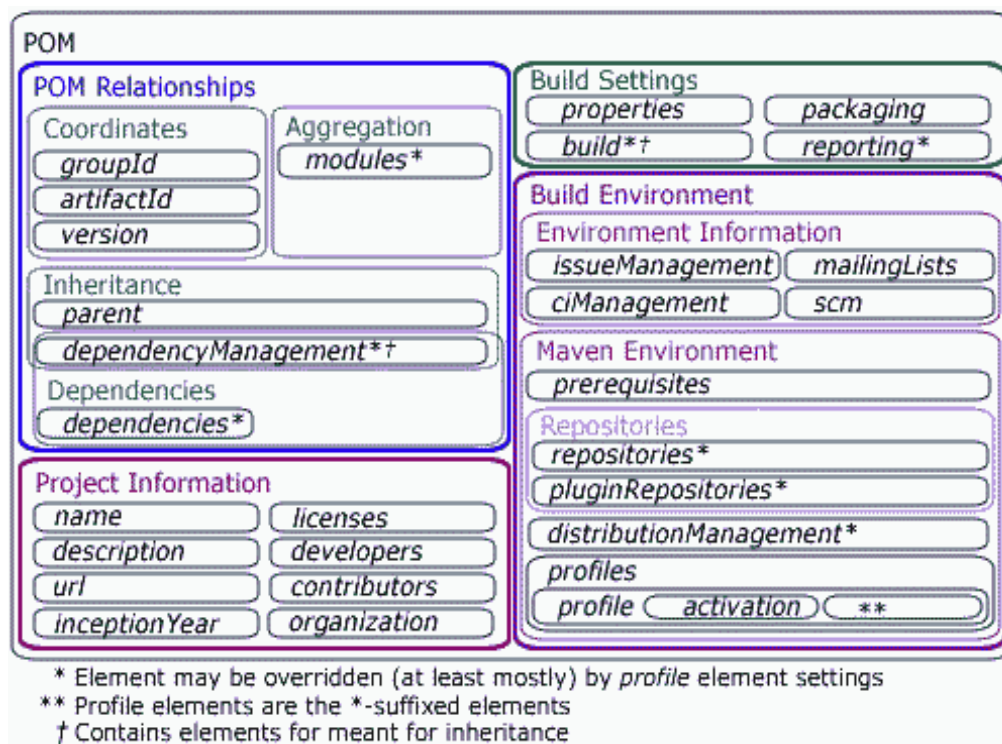
Životní cyklus Mavenu obsahuje několik částí, resp. cílů, které můžeme zadávat k vykonání:

- `compile` – přeloží zdrojové soubory projektu
- `test` – spustí JUnit testy napsané pro projekt
- `package` – zabalí aplikaci do balíčku
- `install` – přidá artefakty do lokální repository
- `deploy` – nasadí výsledné artefakty do vzdálené repository

Spuštění každého cíle musí předcházet vykonání předchozího, pokud ještě nebyl úspěšně vykonaný nebo byly provedeny změny v projektu. Když spustíme některý příkaz, Maven zkontroluje zda předchozí cíl proběhl úspěšně, a pokud ne, spustí jej. Mezi další cíle například patří `clean`, `assembly:assembly`, `site`, `site:deploy` a další.

### 3.2.3 POM

Maven používá ke své činnosti model POM (*Project Object Model*), sloužící k popisu sestavovaného projektu. Je ve formě XML dokumentu, nazvaného `pom.xml`. Maven hledá tento soubor při vykonávání stanoveného cíle, kde nalezne informace potřebné pro jeho splnění. Lze definovat Super POM, který je předkem POMů z jiných projektů a lze tak zavést dědičnost, která umožní vykonávat cíle přes více projektů. POM můžeme rozdělit do čtyř logických celků: POM vztahy, informace o projektu, nastavení sestavování a prostředí sestavení.



Obr. 3.3 Přehled POM modelu, zdroj: [13]

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>MarathoN</groupId>
  <artifactId>Servlets</artifactId>
  <name>MarathoN Servlets module</name>
  <version>0.0.1-SNAPSHOT</version>
  <description>Data collecting servlets for
MarathoN</description>
  <packaging>war</packaging>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.4</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>servlet-api</artifactId>
```

```

        <version>2.5</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring</artifactId>
        <version>2.5.3</version>
    </dependency>
    . . . <!--Další závislosti na knihovnách třetích stran-->
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>2.0.2</version>
            <configuration>
                <source>1.6</source>
                <target>1.6</target>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.mortbay.jetty</groupId>
            <artifactId>maven-jetty-plugin</artifactId>
        </plugin>
    </plugins>
    <resources>
        <resource>
            <directory>src/main/java</directory>
            <includes>
                <include>oscache.properties</include>
            </includes>
        </resource>
    </resources>
</build>
</project>

```

Obr. 3.4 Ukázka POM souboru z projektu MarathoN.Servlets

Mezi výhody, které Maven přináší patří:

- vede vývojáře organizace k dobrým návykům při sestavování projektu
- zahrnuje v sebe komplexní správu projektu s propracovaným systémem dynamického připojování knihoven
- integraci do vývojových prostředí, momentálně jsou podporovány prostředí Eclipse, NetBeans a IntelliJ

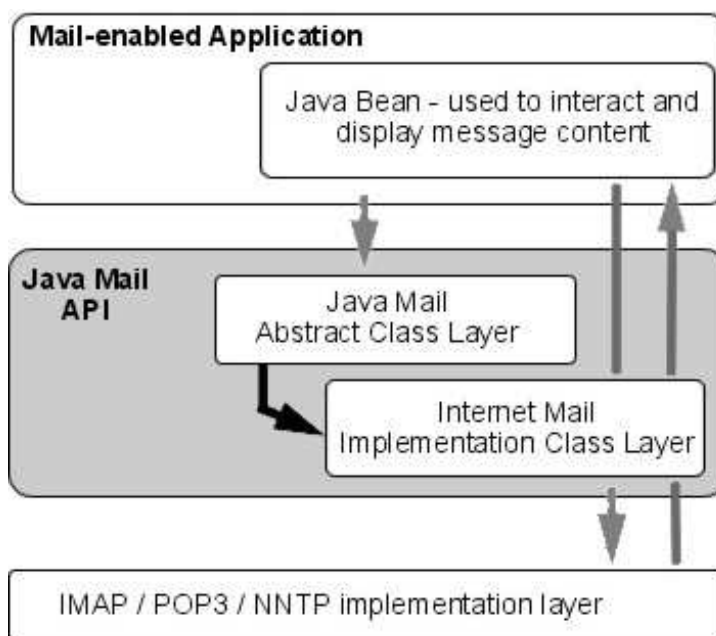
### 3.3 Java Mail

Java Mail poskytuje sadu tříd pro práci s elektronickou poštou na platformě Java. Je součástí Java EE platformy. Java Mail API definuje třídy pro rozesílání a přijímání, jakými jsou Message, Store, Transport, MimeMessage a další. API je rozděleno na abstraktní třídy pro práci s poštou a konkrétní implementaci poskytovatelů jednotlivých protokolů. Java Mail API v sebe zahrnuje nejpoužívanější protokoly jako SMTP, IMAP a POP3. Celková dokumentace JavaMail API je uvedena v [8].

#### 3.3.1 Architektura

Architektura Java Mail API je tvořena v podstatě třemi vrstvami:

- Abstraktní vrstva - deklarující třídy, rozhraní a abstraktní metody s emailovými funkcemi. Třídy jsou nezávislé na komunikačním protokolu.
- Internetová implementační vrstva – implementuje části abstraktní vrstvy dodržením internetových standardů RFC822 (specifikace je uvedena v [12]), MIME a tím implementuje komunikační protokoly.
- Implementační vrstva pro práci s daty – představuje konkrétní práci s daty, tj. manipulaci s částmi emailů, složkami atd. Java používá JAF (*Java Activation Framework*) k zaobalení dat zprávy a zpracování příkazů pro práci s nimi.



Obr. 3.5 Architektura Java Mail

Mezi výhody Java Mail patří:

- standardní API v Javě, zahrnuté v Java EE platformě
- vrstvená architektura a snadné použití

## 3.4 Hibernate

### 3.4.1 Co je Hibernate?

Hibernate je Java framework, zajišťující objektově-relační mapování (*ORM Object Relational Mapping*) a dotazovací služby. Hibernate framework mapuje objektově-orientovaný doménový model do relační databáze. Povoluje vyvíjet třídy k persistenci, dodržující objektově orientované principy jako asociace, dědění, polymorfismus, skládání a kolekce. Hibernate poskytuje pro psaní vlastních dotazů rozšíření SQL jazyka HQL, lze psát i v původním SQL nebo můžeme použít objektově orientované Criteria a Example API. Hibernate je *open source* projekt, vytvořený společností JBoss. Velmi dobře je problematika perzistence v jazyce Java pomocí nástroje Hibernate popsána v [5].

### 3.4.2 Architektura

Hibernate poskytuje jednoduché API, které umožňuje mapování objektů do 20 podporovaných relačních databází. Architekturu vytváří několik prvků, které jsou pro Hibernate specifické:

- transparentní a automatická persistence – zachovává transparentnost doménového modelu vůči persistentnímu mechanismu a obstarává práci s JDBC konektorem. Transparentní je v tom smyslu, že se nevyžaduje, aby doménové třídy dědily ze speciální třídy a je s nimi možné zacházet i mimo kontext perzistence, např. v testech.
- využití POJO objektů – doménový model je realizován POJO objekty (*Plain Old Java Object*), deklarující byznys metody, definující chování a „properties“, definující stav či asociaci mezi objekty
- členitý objektový model – podpora vysoce členěných objektů, které jsou charakteristickým rysem rozsáhlých systémů s mnoha doménovými objekty. Ilustrujícím příkladem může být tabulka Zákazník s atributy domácí adresy a platební adresy, naproti tomu v objektovém modelu může mít třída Zákazník zakomponovanou třídu adresa, která může být typu domácí a platební.
- databázová identita – existují dva prostředky identity, „property“ identifikátor persistentní instance, který je primárním klíčem nebo hodnota vrácena metodou `Session.getIdentifier(Object o)`

### 3.4.1 Výhody a nevýhody

Mezi výhody, které přináší Hibernate a objektově relační mapování, patří:

- snížení času, stráveného psaním kódu, obstarávající perzistenci
- podpora mnoha databází v závislosti na JDBC ovladači, lze jednoduše vyměnit databázi



U Hibernatu jako ORM nástroje musíme počítat se zhoršenou časovou náročností pro přístup k datům. Tato nevýhoda může být v některých aplikacích i kritická k použití.

## 3.5 Wicket

### 3.5.1 Co je Wicket?

Pro platformu Java existuje v dnešní době velké množství webových frameworků a Wicket je jedním z nich. Proč tedy vznikl další webový framework, když jich existuje mnoho a jaké možnosti nabízí si nastíníme v této části. Wicket je open source komponentně orientovaný webový aplikační rámec pro jazyk Java. Komponentová architektura a událostně založený model je podobný přístupu, použitému v knihovně Java Swing. Vývoj internetových a desktopových aplikací jsou dva poměrně odlišné přístupy. Wicket je příkladem nástroje, který se inspiroval ve vývoji desktopových aplikací. Wicket odděluje HTML kód od Java kódu, což následně znamená, že weboví návrháři mohou upravovat HTML stránky pomocí standardních nástrojů jako Dreamweaver, MS FrontPage či jakýkoliv jiný HTML editor. Většina Java frameworků přidává do HTML speciální značky a syntaxi. Wicket nepřidává žádnou speciální syntaxi do HTML, pouze je potřeba označit některé HTML značky, používané Wicketem. Označení je zajištěno id atributem se jmenným prostorem Wicketu `wicket:id`, což vyhovuje XHTML standardu. Bližší informace o práci s Wicket frameworkem lze nalézt v [6] a [7].

### 3.5.2 Vlastnosti

Vlastnosti celého frameworku lze rozdělit do několika oblastí.

- **Jednoduchost**
  - založený na POJO objektech
  - s veškerým kódem a komponentami se pracuje ve stylu Java Swing
  - odpadá nutnost použití mnoha XML konfiguračních souborů
  - řeší problém efektu tlačítka zpět v prohlížeči, a to tak, že si v paměti uchovává předchozí verze stavů komponent
  - lze jednoduše vytvořit stránky formou záložek
  - typová bezpečnost a diagnostika problémů během překladu i za běhu aplikace
  - minimální závislost na zvláštních nástrojích
  - konzistentní komponenty, jejich kontejnery a konvence jejich použití
- **Znovupoužitelnost**
  - napsané komponenty mohou být plně znovupoužitelné
  - znovupoužitelné komponenty mohou být distribuovány v běžných JAR archivech
- **Nerušivý přístup návrhu**
  - HTML návrh neobsahuje programovou sémantiku
  - v návrhu je pouze jeden značkovací konstrukt
  - HTML kód je kompatibilní s jakýmkoliv HTML editorem

- jednoduché zpětné přidání značek Wicketu do HTML kódu, pokud se náhodou smažou
- **Bezpečnost**
  - architektura Wicketu je navržena, tak aby výsledné aplikace splňovaly bezpečný přístup
  - pouze explicitní odkazy mohou odkrýt stav stránek a URL
  - zajištěna typová bezpečnost
- **Efektivita a škálovatelnost**
  - efektivně navržený framework pro jednoduché použití
  - podpora „clusteringu“ pomocí `sessions`

### 3.5.3 Zhodnocení

#### Výhody

- odděluje HTML a Java kód, výsledkem je pak „čistě“ HTML
- práce s komponenty je podobná knihovně Swing, tyto komponenty je možné dědit
- řeší problém z nepěknými URL, tzn. dlouhé URL se pak zobrazují jako jednoduché
- podporuje integraci s AJAX komponentami
- velké množství ukázkových příkladů
- podporuje validaci formulářů a lokalizaci
- a další

#### Nevýhody

- pro webové vývojáře nemusí být blízký komponentový přístup ve stylu Java Swing
- v porovnání s např. JSF obsahuje menší počet komponent
- komponentní architektura, může zpříčinit pomalejší chod aplikace oproti požadavkem řízených webových frameworků
- neexistuje rozumné řešení webového návrhu, Wicket zřejmě neřeší to, že by webová stránka mohla být založena na několika na sebe relativně nezávislých generovaných částí

## 3.6 Hessian a Burlap

Tato kapitola čerpá informace z dokumentace společnosti Caucho [11]. Hessian a Burlap jsou protokoly webových služeb, které umožňují jejich použití bez nutnosti aplikace velkého frameworku a bez dalších podpůrných protokolů. Více než o protokoly, které jsou známé z klasických webových služeb, se jedná spíše o protokoly, umožňující vzdálené volání metod. Burlap je protokol z projektu Caucho založený na XML a Hessian je binární protokol, odvozen od Burlapu. Jelikož je Hessian binární protokol, lze jej dobře použít pro posílání binárních dat bez nutnosti rozšíření protokolu přílohami. Hessian byl vyvinut společností Caucho Technology pod open source licencí (*Apache license*). Jeho implementace existuje do mnoha jazyků jako Java, Python, Action Skript, Adobe Flash a implementace třetích stran C++, .NET C#, PHP, Ruby, Objective-C, D a Enlang.

Vytvoření Hessian služby v Javě je zapotřebí čtyř kroků (pro službu využívající protokol Burlap je postup obdobný):

- Vytvoření Java rozhraní jako veřejného API

- Vytvoření klienta využívající HessianProxyFactory
- Vytvoření třídy implementující službu
- Konfigurace služby v servletovém stroji

Jak tedy vypadá systém využívající Hessian protokol?

### Klient

Volání Hessian služby z Java klienta je jako volání metody. HessianProxyFactory vytváří proxy služby jako normální objekt s možností vyvolání výjimky v případě přerušeného vzdáleného spojení. Každá služba dodržuje standardní Java rozhraní, popisující službu. Můžeme použít jakékoliv Java typy i objekty, protože Hessian podporuje jejich serializaci.

```
String url = "http://hessian.caucho.com/marathon/service";

HessianProxyFactory factory = new HessianProxyFactory();
ServiceAPI service = (ServiceAPI) factory.create(ServiceAPI.class,
url);

System.out.println("Service: " + service.getServiceName());
```

Obr. 3.6 Ukázka implementace Hessian klienta

### Služba

Klient pomocí HessianProxyFactory a specifikovaného URL vytvoří zástupný objekt služby. Služba pouze implementuje své rozhraní. Hessian knihovna umožňuje vytvořit službu zděděním z třídy HessianServlet.

```
public interface ServiceAPI {
    public String getServiceName();
}
```

Obr. 3.7 Ukázka rozhraní Hessian služby

```
public class MarathonService extends HessianServlet implements
ServiceAPI {
    private String _serviceName = "Marathon Service";

    public void setServiceName(String serviceName)
    {
        _serviceName = serviceName;
    }

    public String getServiceName()
```

```
{  
    return _ serviceName;  
}  
}
```

Obr. 3.8 Ukázka implementace Hessian služby

### **Vlastnosti**

Hessian i Burlap se vyznačují následujícími vlastnostmi:

- jednoduchý nenáročný protokol, jako transportní vrstva se využívá HTTP
- odpadá nutnost zdoluhavé implementace služeb, používajících SOAP
- používán primárně pro jednoduché klienty a mobilní zařízení

Tyto protokoly lze s výhodou využít, pokud máme všechny komunikující systémy pod kontrolou a navíc se jedná o Java aplikace a to zejména kvůli jejich jednoduchosti a rychlosti. Pokud systémy nemáme pod kontrolou, které navíc komunikují mezi různými platformami jsou k použití výhodnější klasické webové služby.

# 4 Řešení systému pro cílený e-mail marketing

## 4.1 Úvod

V předchozích kapitolách jsme se seznámili s teoretickým pozadím e-mail marketingu, se zákony, upravující jeho použití a s některými technologiemi, použitými při vývoji projektu. Další kapitoly popisují samotné řešení systému. Co tedy iniciovalo myšlenku vzniku nového systému, který bude zajišťovat služby e-mail marketingu? Především to byl záměr vybudovat samostatný systém, poskytující základ pro tuto obchodní činnost. Systém je vyvíjen na základě požadavků reklamní a marketingové společnosti Triax CZ s.r.o.

E-mail marketing je v dnešní době rozšířeným reklamním a obchodním nástrojem. Jistě tedy každý z nás přišel do styku s e-maily, které byly jeho prostřednictvím odeslány. V podstatě každá firma potřebuje své zákazníky oslovit a udržet s nimi kontakt, k tomu jí internet a e-mail marketing nabízí příležitost k přímé komunikaci a k zaznamenávání zpětných vazeb. Firmy mohou využít služeb společností, nabízející komplexní služby e-mail marketingu. Mezi tyto služby patří poskytnutí softwarového systému, který je většinou přístupný přes webové rozhraní, dále je nutné poskytnout nezbytné serverové systémy, hardware a poradenské služby. Nabídky těchto společností si jsou velmi podobné a liší se především počtem nabízených funkcí, kvalitou zpracování a cenou.

## 4.2 Analýza a návrh

### 4.2.1 Požadavky na systém

Výsledný systém pro cílený e-mail marketing nalezne uplatnění v menší reklamní společnosti, která pak s jeho pomocí bude poskytovat e-mail marketingové služby svým klientům. Požadavky na systém byly vesměs přímočaré. Výsledný systém bude umožňovat komplexní správu e-mail marketingových kampaní. Pomocí něj bude možné vytvořit HTML dokument, který je součástí kampaně. Ten lze rozeslat na předem specifikovaný seznam odběratelů. Systém pak bude zaznamenávat odezvy jednotlivých odběratelů a vyhodnocovat úspěšnost jednotlivých marketingových kampaní. Klient, který si danou kampaň objednal, ji pak může průběžně sledovat. Všechny požadavky lze shrnout do následujícího seznamu:

#### Funkční požadavky

- zadávání a správa kampaní
- správa odběratelů a jejich strukturování do adresářů
- odesílání e-mailů specifikovaným odběratelům
- statistické vyhodnocování nasbíraných dat v rámci kampaně

#### Nefunkční požadavky

- přívětivé grafické uživatelské rozhraní s pohodlným ovládáním

- architektura systému by měla být snadno rozšiřitelná

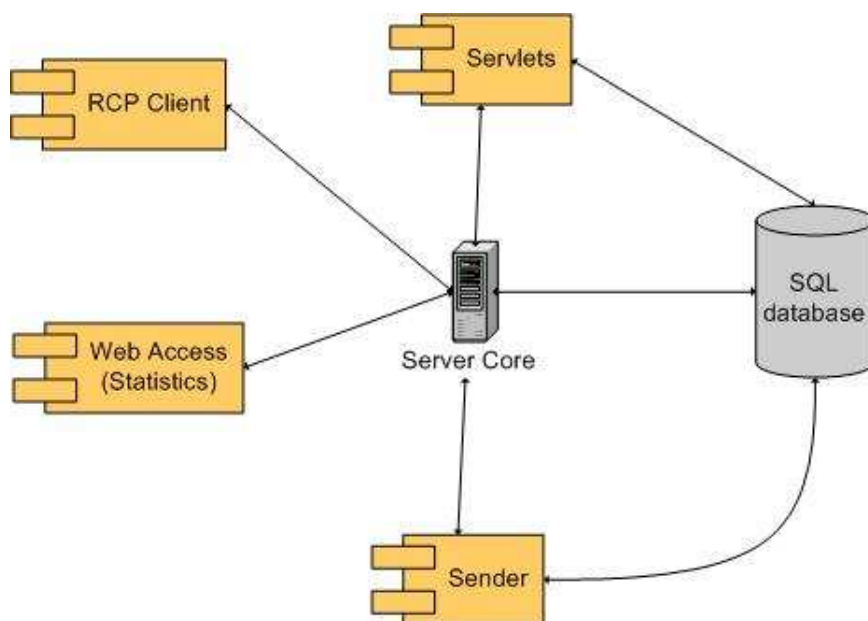
#### 4.2.2 Funkční celky

Systém se skládá ze dvou hlavních částí - klientské a serverové. Klientská část je realizována formou tlustého klienta a plní funkci administrační konzole pro správu kampaní, odběratelů, adresářů a dalších. Serverová část se skládá ze služby poskytující aplikační logiku, modulu pro rozesílání e-mailů, servletové části, sloužící pro sběr dat, a z webového rozhraní, zpřístupňující jednotlivé statistiky marketingových kampaní.

#### Architektura modulů

Systém je rozdělen do čtyř navzájem komunikujících modulů. Každý z těchto modulů je samostatným projektem s vazbami na ostatní.

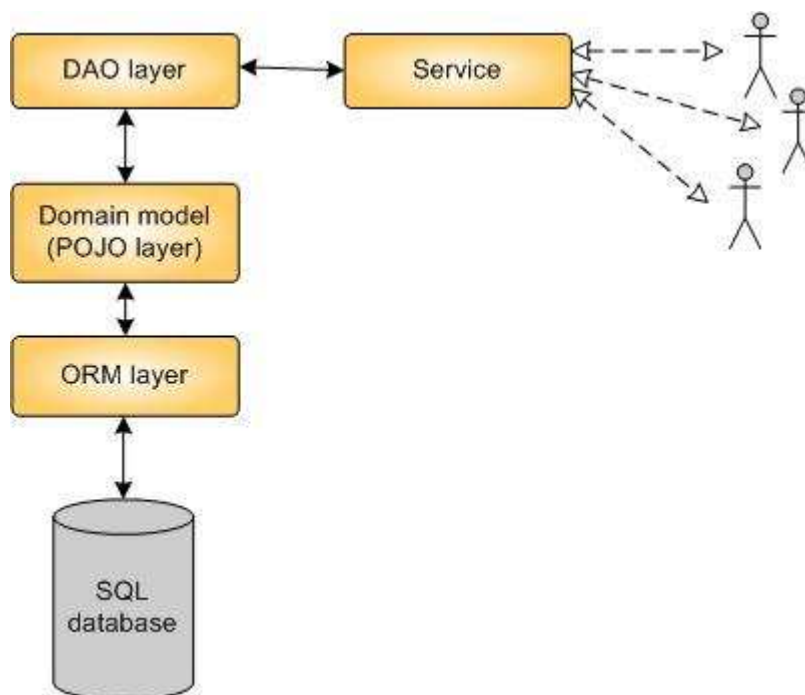
- RCP klient – administrační konzole pro správu kampaní
- Serverový modul – část, obsahující aplikační logiku, doménový model a část pro rozesílání
- Servletový modul – obsahuje sadu servletů, obstarávajících sběr dat od uživatelů a poskytujících data e-mailům
- Webová část – webová aplikace zpřístupňující statistiky klientům



Obr. 4.1 Hlavní moduly systému

### 4.2.3 Serverový modul

Modul serveru v sobě sdružuje serverovou logiku a modul pro rozesílání e-mailů. Serverovou logikou se myslí část systému, obsahující webovou službu, která nabízí rozhraní s serverovými funkcemi. Architektura této části je na obrázku č. 4.2.



Obr. 4.2 Architektura serveru

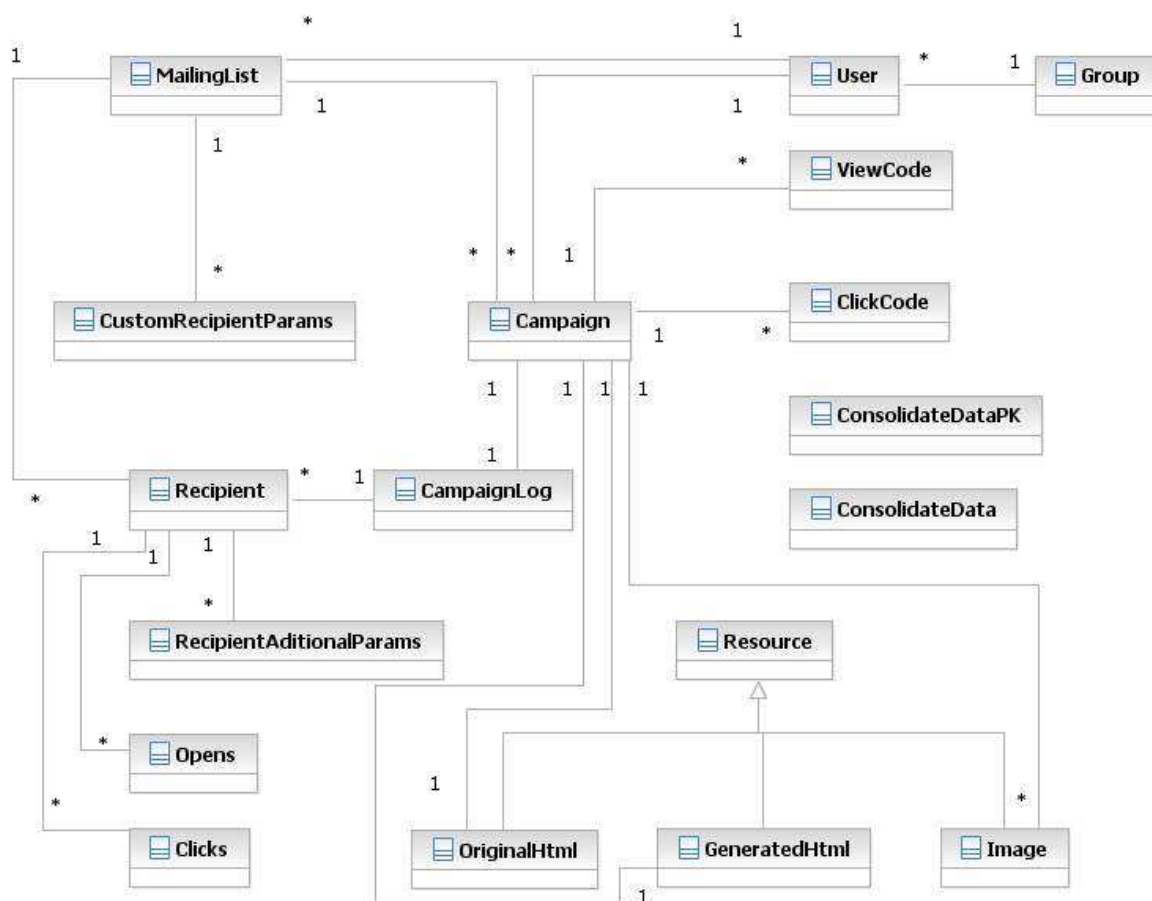
Pro mapování doménového modelu do relační databáze je využit ORM nástroj Hibernate, viz kapitola 3.4. Nad doménovou vrstvou s POJO objekty je vrstva s DAO objekty, kde DAO (*Data Access Object*) poskytuje rozhraní persistentního mechanismu, zajišťuje provádění specifických operací a skrývá databázové detaily. Obsahuje CRUD (*Create Update Delete*) operace s entitami a zaobaluje v sebe různé dotazy do databáze.

Na pomyslném vrcholu architektury serveru stojí služba, poskytující svým rozhranním funkce, které mohou být volány potencionálními klienty služby. V její implementaci je zakomponován návrhový vzor příkaz, který zaobaluje příkazy pro ukládání, editaci a mazání jednotlivých entit a entit na nich závislých. Další částí, která je implementována na straně serveru, je modul pro rozesílání e-mailů. Jeho spuštění je dostupné z metod, které specifikuje rozhraní služby.

#### Doménový model

Na obrázku č. 4.3 je zobrazen doménový model systému, který je mapován do relační databáze. Primárním byznys objektem je Campaign, který obsahuje všechny vlastnosti e-mail marketingové kampaně. Může být rozeslána pouze na jeden adresář MailingList. Ten obsahuje seznam odběratelů Recipients, kteří se přihlásili k odběru reklamní pošty. MailingList obsahuje i

seznam atributů – dodatečných vlastností CustomRecipientParams, které mohou mít odběratelé v daném adresáři. Tyto vlastnosti mohou mít jednotliví odběratelé vyplněny. Při odesílání kampaně na konkrétní adresář se pak mohou tyto vlastnosti využít při personalizaci e-mailu. Pokud byla kampaň označena jako personalizovaná, pak se během zpracování e-mailu k odeslání nahradí značky - vzory, označující vlastnosti odběratele a speciální vlastnosti, přidané adresářem, za hodnoty vlastností konkrétního odběratele. CampaignLog slouží k ukládání informací o času a stavu odeslání kampaně konkrétnímu odběrateli. Kampaň obsahuje dva druhy HTML souborů: původní a zpracovaný k odeslání. Původní HTML soubor představuje originální HTML dokument, navržený designérem kampaně, který obsahuje odkazy, obrázky a případné personalizované značky ke zpracování. Zpracovaný HTML dokument prošel procesem zpracování a označené odkazy a zdroje obrázků jsou nahrazeny adresami serverů, které je budou zpracovávat. Kampaň dále může obsahovat sadu obrázků, použitých v e-mailu, její součástí jsou také vygenerované kódy kliků a kódy otevření. Další dvě entity Opens a Clicks jsou určeny pro sběr dat z rozeslaných e-mailů. Model také obsahuje jednoduchý systém uživatelů User klientského systému webových statistik, kteří jsou strukturováni do skupin Group.



Obr. 4.3 Doménový model systému



## Modul pro rozesílání e-mailů

Součástí serveru je také modul, starající se o rozesílání e-mailů. Ten na vstupu obdrží marketingovou kampaň se zpracovaným HTML dokumentem. Pokud je kampaň označena jako personalizovaná, zpracuje se HTML dokument ještě jednou a do e-mailu se vloží personální informace odběratele. Jestliže kampaň není personalizovaná odešle se HTML e-mail přímo odběrateli bez dalšího zpracování. Modul pro rozesílání využívá standardního frameworku pro práci s elektronickou poštou JavaMail.

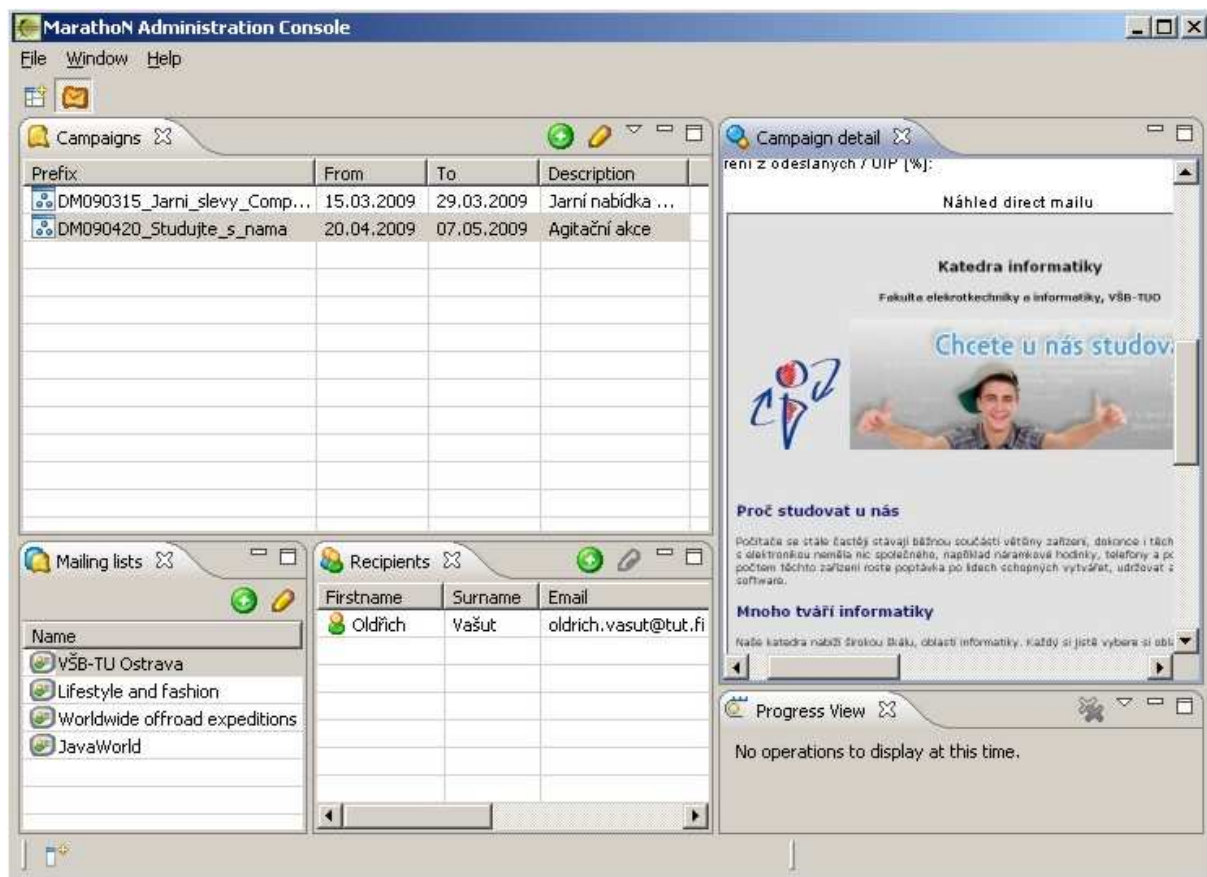
### 4.2.4 RCP klient

RCP klient plní funkci administrační konzole pro zadávání a správu kampaní, adresářů, odběratelů a dalších. RCP klient je naimplementován na platformě Eclipse RCP viz kapitola 3.1. Administrační rozhraní formou tlustého klienta Eclipse RCP byla zvolena proto, že přináší vyšší uživatelský komfort při práci se systémem a lze těžit výhody z její architektury. Klient obsahuje několik pohledů, perspektiv a průvodců.

#### Pohledy (*Views*)

Pro prezentaci uložených dat je v RCP klientu využito Eclipse pohledů, z nichž většina obsahuje komponentu s tabulkou, která zobrazuje data. Klient se skládá z následujících pohledů:

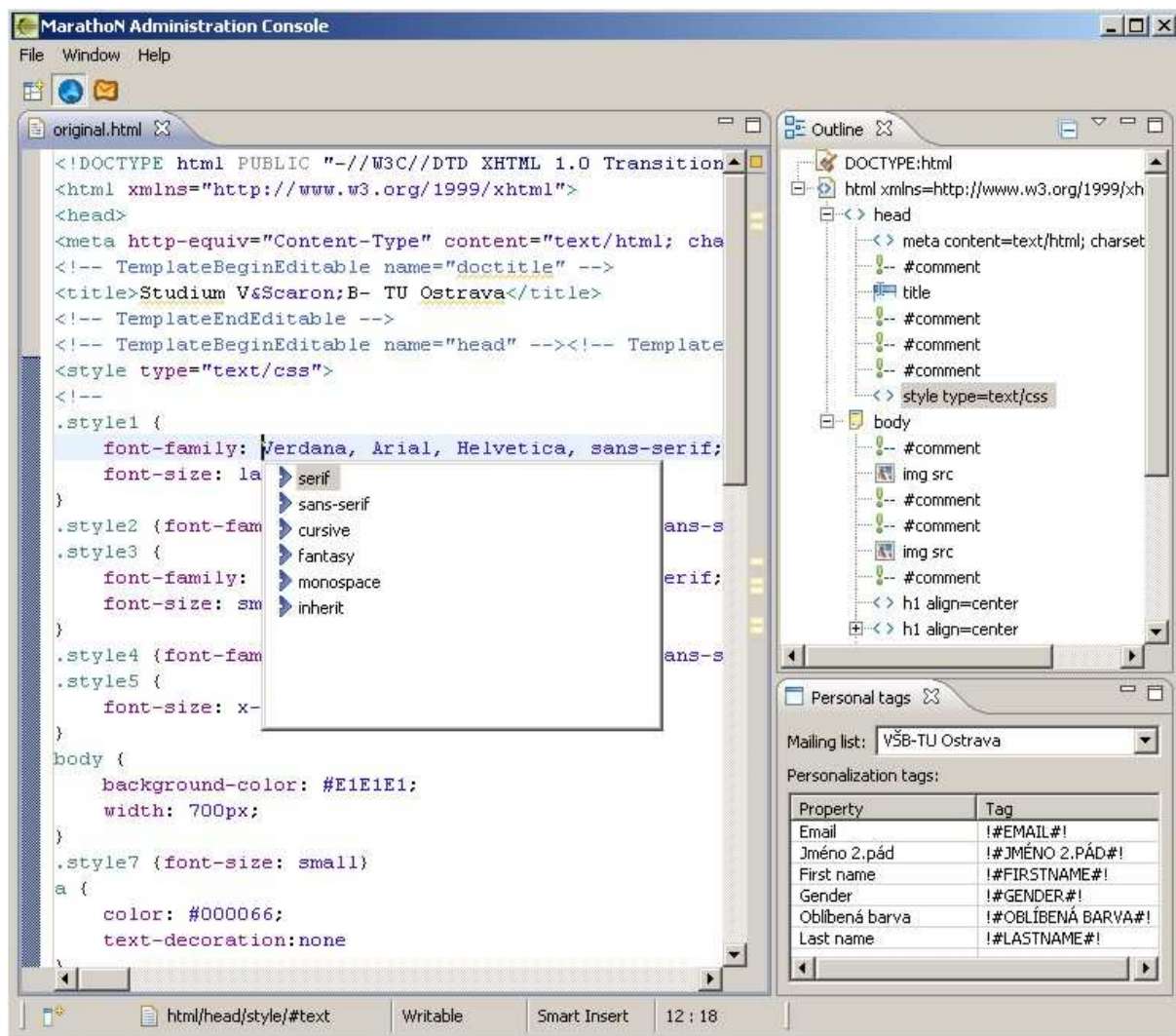
- **Kampaně (*Campaigns*)** – obsahuje seznam vytvořených kampaní a poskytuje možnosti pro vytvoření a editaci kampaně a její odeslání
- **Detail kampaně (*Campaign Detail*)** – poskytuje podrobné informace o kampani
- **Adresáře (*mailing lists*)** – v této části lze spravovat adresáře a je zde možnost importu odběratelů z csv souboru. Tato funkcionality je zde pro případ, že firma si zakoupí (či jinak legálně získá) seznam zákazníků s emailovými adresami a s dostatečným povolením k příjmu nabídek.
- **Odběratelé (*Recipients*)** – zajišťuje správu odběratelů v jednotlivých adresářích
- **Skupiny (*Groups*)** – poskytuje správu uživatelských skupin systému
- **Uživatelé (*Users*)** – seznam uživatelů majících přístup ke statistikám kampaní



Obr. 4.4 Administrační konzole - Správa kampaní

## Editory (Editors)

Jelikož platforma Eclipse RCP nabízí mnoho již vytvořených komponent, lze jich s výhodou ve vyvíjených aplikacích využít. Pro editaci vstupního HTML dokumentu jsem využil HTML editor, který je integrován v projektu Eclipse IDE Web Tools Platform. Nabízí barevné zvýraznění syntaxe kódu a kontextové našeptávání. V tomto editoru je možné celý HTML soubor napsat a navrhout, nicméně se předpokládá že se pro předpřípravení a navržení HTML souboru bude používat profesionální editor třetích stran jako je například Macromedia Dreamweaver.



Obr. 4.5 Administrační konzole – HTML editor

### Perspektivy (*Perspectives*)

Použití více perspektiv v klientu je velmi výhodné, neboť perspektiva sdružuje sadu pohledů, editorů a ovládacích prvků. Tímto nastavením uživatelského prostředí vytvoří pracovní sestavu nástrojů pro společné činnosti, které chce uživatel provádět. Práce uživatele je rozdělena do tří perspektiv.

- Správa kampaní – sdružuje v sebe pohledy pro správu kampaní, adresářů a odběratelů
- Návrh e-mailu – obsahuje HTML editor s Outline pohledem a nástrojem pro získání personalizovaných značek, představující jednotlivé vlastnosti z adresáře
- Správa uživatelů – obsahuje pohledy pro správu uživatelů, kteří mohou být strukturováni do skupin

## Průvodci a dialogy (*Wizards and Dialogs*)

Pro snadné vkládání a editaci údajů jednotlivých položek v tabulkách je využito JFace průvodců, které nabízí Eclipse RCP. Práce s těmito průvodci není zvlášť náročná a poskytuje jednoduché ovládání. Ukázka práce s průvodcem je zachycena na obr. 4.6. Systém rovněž zahrnuje množství dialogových oken sloužící k dotazovacím, informativním, varovným a chybovým zprávám.

Property	Value
Jméno 2.pád	Oldřichu
Oblíbená barva	modrá

Obr. 4.6 Průvodce založením nového odběratele

Would you like to send campaign "DM090420\_Studujte\_s\_nama" to recipients of mailing list "VŠB-TU Ostrava"?

Obr. 4.7 Dialogové okno s dotazem na rozeslání reklamních e-mailů

#### 4.2.5 Servletový modul

Tato část v sebe sdružuje sadu servletů, které jsou spuštěny na aplikačním serveru resp. v servletovém kontejneru. Modul obsahuje čtyři druhy servletů, starajících se o sběr a poskytování dat odeslaným e-mailům v průběhu kampaně. Všechny servlety přistupují do databáze přes JDBC driver, bez využití ORM nástroje Hibernate. Přímý přístup do databáze, bez použití mapovacího nástroje, je zvolen kvůli jeho rychlosti. Po rozeslání e-mailů na velký počet uživatelů může být frekvence požadavků do databáze během krátkého období příliš velká. Práci s e-maily marketingové kampaně obstarávají následující typy servletů:

- `DMClick servlet` – zaznamenává do databáze uživatelův klik (obsahující také IP adresu a čas kliku) na sledovaný odkaz a přesměruje jej na původní adresu odkazu
- `DMOpen servlet` – zaznamenává do databáze otevření e-mailu (zaznamenává se IP adresa uživatele a čas otevření)
- `Image servlet` – poskytuje obrázky z databáze odeslaným e-mailům
- `Remove servlet` – na základě kliku na odkaz, který znamená požadavek na odebrání uživatele ze seznamu odběratelů nabídek, jej odebere ze seznamu. V podstatě to znamená že se nastaví vlastnost odběratele `unsubscribe date` na aktuální datum při provedení kliku. Uživatelé již pak nejsou zasílány další nabídky, určené pro adresář, v němž byl odběratel zapsán. E-maily, které tento odkaz obsahují, pak dodržují ustanovení OPT-OUT viz kapitola 2.8.3.

#### 4.2.6 Webové statistiky

Tento modul poskytuje webové rozhraní pro prohlížení výsledných statistik e-mail marketingových kampaní. Pro jeho implementaci byl využit webový framework Wicket viz kapitola 3.5. Modul je určen pro klienty resp. objednatele marketingové kampaně. Po přihlášení klienty do systému jsou zde připraveny tři druhy jeho kampaní, rozdělených do skupin podle doby trvání. První jsou aktuální kampaně, které již byly rozeslány a jejich doba platnosti stále pokračuje. Ukončené kampaně jsou ty, které již proběhly. Dále zde jsou připravované kampaně, které budou rozeslány v budoucnu. Poslední skupinou jsou všechny kampaně, které proběhly, probíhají a budou probíhat. Jednotlivé kategorie nabízejí seznam e-mail marketingových kampaní s možností zobrazení jejich vypočítanými statistickými daty v tabulkách. Každá z těchto zobrazených kampaní nabízí tři druhy statistik.

##### Souhrnné statistiky po dnech

Obsahuje souhrnné statistiky za jednotlivé dny trvání kampaně i souhrnné za všechny dny. Ve sloupcích tabulky jsou vypočítány pro jednotlivé dny následující typy hodnot:

- Počet zobrazení – celkový počet zobrazení e-mailu kampaně za konkrétní den
- Počet zobrazení / UIP – počet zobrazení e-mailu kampaně za konkrétní den, kde odběratelé jsou započítáni pouze jednou podle stejné IP adresy (UIP - uživatel IP adresa)
- Počet kliků – počet všech kliků v e-mailu od všech odběratelů za konkrétní den

- Počet kliků (%) – podíl počtu kliků konkrétního dne vůči všem klikům za dobu trvání kampaně
- Počet kliků / UIP – počet kliků kampaně za konkrétní den, odběratelé jsou započítáváni pouze jednou podle stejné IP adresy
- CTR (%) – podíl kliků kampaně vůči jejich otevření, udává procentuální úspěšnost kliků z otevřených kampaní tj., kolik odběratelů, kteří si v konkrétní den otevřelo e-mail kampaně, si klikli na některé odkazy
- CTR / UIP (%) – podobně jako procentuální úspěšnost CTR, s tím rozdílem, že odběratelé se stejnou IP adresou jsou započtení pouze jednou

### Statistiky prokliků

Tato část poskytuje statistické vyhodnocení pro každý odkaz (proklik) v e-mailu kampaně. Odkazy jsou označeny číselnými identifikátory, přiřazenými podle identifikačního atributu odkazu `triax:id`.

- Počet zobrazení – počet zobrazení e-mailu kampaně
- Počet kliků – počet všech kliků na daný odkaz v e-mailu kampaně
- Podíl kliků – podíl kliků konkrétního odkazu vůči celkovému počtu kliků e-mailu kampaně
- Počet zobrazení / UIP – počet zobrazení e-mailu kampaně, uživatel je započítán pouze jedenkrát podle stejné IP adresy
- Počet kliků / UIP – počet kliků na konkrétní odkaz, uživatel je započítán pouze jedenkrát podle stejné IP adresy.

### Denní statistiky prokliků

Tato statistika prezentuje počet kliků na jednotlivé odkazy za každý den trvání kampaně, zahrnuje i celkové počty všech kliků za konkrétní den a celkové počty kliků na daný odkaz za celou dobu trvání kampaně.

### Souhrnné statistiky

Souhrnné statistiky zobrazují přehledové statistiky pro jednotlivé kampaně ve společné tabulce. Jsou zde zahrnuty celkové počty otevření e-mailu kampaně, počty kliků na odkazy a počty otevření a kliků, kde je stejný uživatel započten pouze jednou.

## 4.3 Životní cyklus kampaně

Pro osvojení si systému je nutné pochopit princip, na jakém pracuje. Jeho stěžejní částí je životní cyklus kampaně, který je popsán v následujícím textu.

Na obrázku č. 4.8 je schématicky zaznamenán životní cyklus kampaně. Na počátku je nutné vytvořit HTML soubor, který bude tvořit obchodní zprávu, rozeslanou v rámci kampaně. Návrh tohoto dokumentu se řídí několika jednoduchými pravidly tak, aby bylo možné jeho správné zpracování. Pravidla tvoří několik speciálních konstruktů, které musí, případně může zdrojový kód HTML dokumentu obsahovat. Tyto konstrukty jsou:

- `triax:open` – tímto identifikátorem je označen obrázek, který je zpracován pro kontrolu otevření e-mailu

př.: `<img src="" triax:open="true" height="1" width="1"/>`

- `triax:id` – tento atribut HTML značky slouží k označení a identifikaci odkazů, které jsou určeny ke sledování. Z těchto odkazů se následně vypočítávají statistiky. Hodnota tohoto tagu je celé kladné číslo, označující odkaz. V e-mailu se může vyskytovat více odkazů se stejným identifikátorem, což pak znamená, že se jedná o stejný odkaz a statistiky jej budou brát jako jednu položku.

př.: `<a href="http://www.vsb.cz" triax:id="10" target="_blank">`

- `triax:googleanalytics` – tato značka je prostředkem pro integraci kampaně s nástrojem Google Analytics. Kampaň lze pak vyhodnocovat i tímto externím nástrojem.

př.: `<img triax:googleanalytics="true" src="" width="1" height="1">`

- `triax:remove` – tento atribut označuje odkaz, který slouží k odhlášení odběratele z příjmu emailových nabídek

př.: `<a href="" triax:remove="true" target="_blank">`

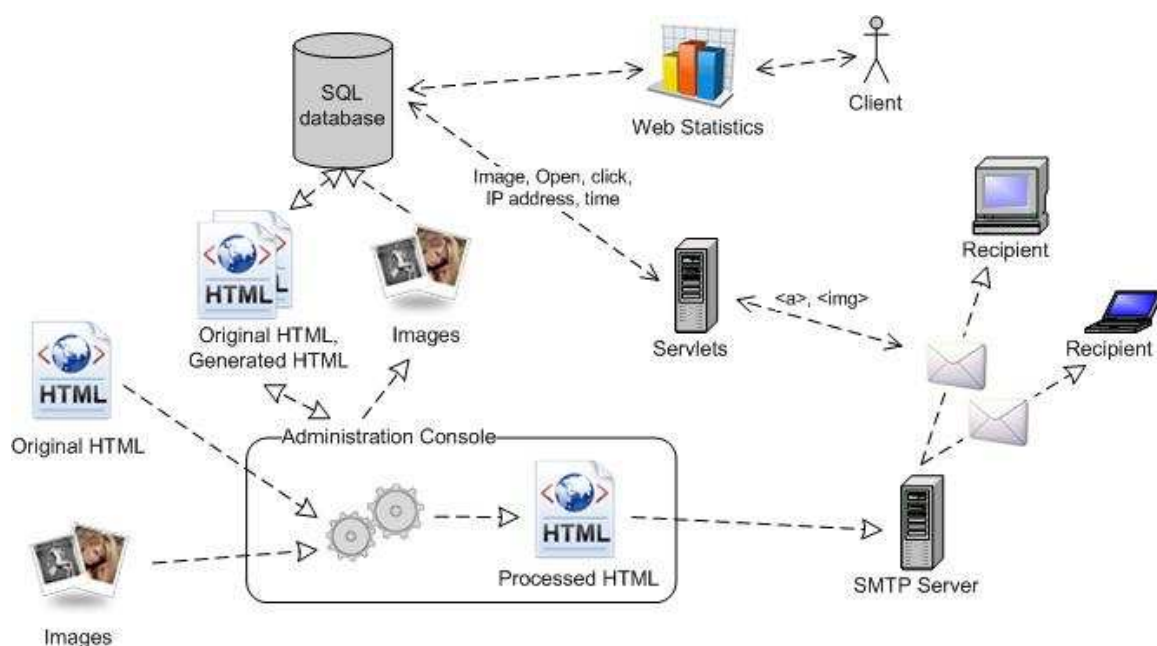
Pokud je kampaň personalizovaná, může e-mail obsahovat vzory, označující vlastnosti odběratelů, které jsou následně při odesílání nahrazeny jejich hodnotami. Příklady takových vzorů ukazuje tabulka č. 4.1.

Značka	Vlastnost	Hodnota
!#EMAIL#!	Email	oldrich.vasut@tut.fi
!#FIRSTNAME#!	Jméno	Oldřich
!#LASTNAME#!	Příjmení	Vašut
!#GENDER#!	Pohlaví	Muž
!#JMÉNO_2PÁD#!	Jméno ve 2.pádu	Oldřichu
!#OBLÍBENÁ_BARVA#!	Oblíbená barva	modrá

Tab. 4.1 Příklad vzorů vlastností odběratele

Navržený HTML dokument je následně vstupem pro vytvoření e-mail marketingové kampaně pomocí Administrační konzole. HTML soubor se zpracuje a systém vygeneruje upravený HTML soubor, oba dva soubory originální i vygenerovaný jsou spolu s obrázky uloženy do databáze. Při zpracování HTML souboru se pracuje se speciálními konstrukty, které byly popsány výše. Tyto konstrukty se zpracovávají následovně. `triax:open` adresa zdroje se nahradí adresou `DMOpen servletu`, zpracovávající požadavek na otevření e-mailu. Otevření e-mailu se tedy pozná podle načtení nepatrného obrázku. `triax:id` adresa odkazu se uloží do databáze a nahradí se adresou `DMClick`

servletu, zpracovávající požadavek na odkaz. Všechny adresy obrázků jsou nahrazeny postupně adresy `getImage` servletu, obstarávající načítání obrázků z databáze. Po doplnění ostatních vlastností kampaně jako je identifikátor, název, období platnosti, adresář, na který se kampaň bude rozesílat, je takto připravená kampaň k odeslání. Dalším krokem je tedy odeslání kampaně odběratelům, specifikovaným vybraným adresářem. Před odesláním e-mailu odběrateli se zkontroluje, zda je kampaň personalizovaná a pokud ano, vymění se v HTML e-mailu všechny nalezené vzory, určující vlastnosti odběratele, hodnotami vlastností odběratele. Jestliže kampaň personalizovaná není, odešle se e-mail beze změn.



Obr. 4.8 Životní cyklus e-mail marketingové kampaně

Druhou část cyklu kampaně tvoří sběr dat z odeslaného e-mailu odběrateli. Po otevření e-mailu a načtení obrázků se zaznamená, že odběratel si otevřel daný e-mail (viz výše). Pokud odběratel klikne na odkaz, zaznamená se jeho klik do databáze a přesměruje se na původní adresu odkazu.

Poslední částí je vyhodnocování nasbíraných dat a jejich prezentace formou statistických tabulek.

## 4.4 Nasazení

### 4.4.1 Model nasazení systému

Systém sestává ze čtyř relativně samostatných modulů, které mohou být nasazeny ve výsledném prostředí různým způsobem. Jak již bylo zmíněno v předchozích kapitolách, systém se skládá z klientské a serverové části. Klientská aplikace Administrační konzole, která je implementována v Eclipse RCP, je instalována na klientském počítači jako samostatná aplikace.



Serverové moduly `Server`, `Servlets` a `Web statistiky` musí být nasazeny na aplikačním serveru. Jelikož jsou to v podstatě samostatné aplikace, může každá pracovat zvlášť na jiném aplikačním serveru nebo mohou být všechny umístěny společně na jednom aplikačním serveru. Jako aplikační server může být použit jakýkoliv servletový kontejner vyhovující specifikaci `Servlet API`, kterými jsou například `Apache Tomcat`, `Mortbay Jetty`, `JBoss` a další.

#### 4.4.2 Instalace a konfigurace

Instalace systému se sestává z několika kroků. Klientská aplikace `Administrační konzole` se instaluje jednoduše, pouhým přepokopírováním do souborového systému na klientském počítači. Na OS `Windows` ji lze spustit pomocí souboru `marathon.exe`. Aplikace `Server`, `Servlets` a `Web Statistiky` jsou servletové resp. webové aplikace, které se nasazují do servletového kontejneru, viz výše.

Pro bezchybnou práci celého systému je důležitá část konfigurace. Každý modul má prostředky, které umožní konfiguraci potřebných parametrů. Následující body ukazují, jaké konfigurační soubory vyžadují jednotlivé moduly.

- `Server` – pro nastavení serveru jsou k dispozici dva soubory. Jeden s konfiguračními údaji o připojení k databázi pro ORM `Hibernate`, který má název `persistence.xml` a druhý `sender.properties` pro nastavení modulu, rozesílajícího e-maily.
- `Servlety` – obsahují konfigurační soubory pro nastavení připojení k databázi `servlets.properties` a soubor k nastavení cache mechanismu `oscahce.properties`
- `Web statistiky` – je třeba nastavit připojení k databázi pomocí konfiguračního souboru `hibernate.cfg.xml`.

#### 4.5 Implementační detaily

V závislosti na návrhu modulů systému je jeho implementace rozdělena do pěti Java projektů:

- `MarathoN.Main` – slouží pro automatické sestavení všech projektů níže uvedených. Obsahuje `SuperPOM`, ze kterého dědí `POMy` jednotlivých projektů, `Maven` pak pomocí něj dokáže sestavit všechny odvozené projekty.
- `MarathoN.RCP` – projekt klientské aplikace v `Eclipse RCP`
- `MarathoN.Server` – sdružuje v sobě třídy aplikační logiky serveru
- `MarathoN.Servlets` – projekt pro modul `servetů`
- `MarathoN.Web` – projekt webové aplikace se statistikami

Celý systém je implementován v jazyce `Java` verze 1.6 a je vyvíjen a testován s těmito verzemi nástrojů:

- `Eclipse RCP 3.2 Europa`
- `Maven 2.0`

- Hibernate 3.3.1
- Wicket 1.2.6
- Jetty Server 6.1
- MySQL 5.1

Veškeré zdrojové kódy, konfigurační a jiné pomocné soubory jsou součástí elektronické přílohy na DVD nosiči. Na tomto médiu se rovněž nachází připravená distribuce systému pro testovací účely. Distribuce obsahuje databázi MySQL s připraveným schématem *marathon*, servletový kontejner Jetty Server s nainstalovanými aplikacemi Server, Servlets a Web. Lze zde nalézt i klienta administrační konzole Eclipse RCP spolu s návodem na spuštění celého systému.

## 5 Závěr

Jsem rád, že jsem mohl v této práci řešit problém ze společensko-technické praxe. Problematika e-mail marketingu je obzvlášť zajímavá a v současném světě obchodu užitečná, atraktivní a společnostmi hojně využívaná. Hlavním cílem této práce bylo navržení a implementace systému pro cílený e-mail marketing, kterého jsem v závěru dosáhl. Výsledný systém je komplexnějšího charakteru a jeho implementace si vyžádala využití několika podpůrných technologií. Použité technologie, jako Maven, Eclipse RCP, Hibernate, Wicket a další, na jedné straně výrazně zjednodušují práci, na druhé straně je však nutné se s těmito nástroji seznámit, prostudovat, jak fungují, a zajistit výslednou integraci, která ústí v konečné řešení. Další cíl, který jsem si kladl na počátku práce, byl seznámit čtenáře přehledově s použitými technologiemi. Především tím způsobem, aby si o nich udělal představu, pochopil proč jsem konkrétní nástroje použil, jaké výhody přinesly při implementaci projektu, či naopak jaké nevýhody lze od nich očekávat. Při řešení práce jsem mohl zužitkovat znalosti z předchozích projektů, které většinou využívaly především Eclipse RCP, Hibernate či Wicket. Během vývoje projektu jsem si pak mohl v těchto oblastech rozšiřovat znalosti. Výsledný systém poskytuje podporu pro provozování cíleného e-mail marketingu, který se sice nevyrovnává produktům společností zajišťujících komplexní e-mail marketingové služby. Nicméně systém je dobrým základem pro provoz takové služby a pomocí navržené modulární architektury jej lze dobře rozšiřovat.

Jelikož tato práce byla časově omezená, systém zahrnuje funkce, které byly na počátku specifikovány a funkce s nimi související. Další vývoj projektu bude následně záviset na jeho využívání a požadavcích reklamní společnosti, pro kterou je systém určen. V zásadě by se budoucí rozvoj projektu dal rozdělit do několika etap. Následující vlastnosti a vylepšení jsou nad rámec diplomové práce. Mohou však v budoucnu zkvalitnit práci se systémem a zvýšit jeho užitnou hodnotu. V závislosti na velikosti provozu a frekvenci požadavků do databáze od odběratelů bude třeba sledovat rychlost odezvy servletového modulu. Modul servletů sice obsahuje „cache“ sql dotazů, nicméně zátěžové testy budou nutností. V budoucnosti by k části pro rozesílání e-mailů mohl vzniknout modul pro plánování aktivit, který by např. spouštěl rozesílání kampaní ve specifikovanou dobu a později by tento modul mohl být využit i ke spouštění jiných činností. Administrační konzole může být vylepšena o funkcionality jako například HTML editor s doplněním o WYSIWYG editor s mechanismem šablon, umožňující pohodlnější návrh e-mailů. Import kontaktů ze souboru do adresáře může být doplněn o nové formáty souborů jako \*.xls a seznam kontaktů z poštovních klientů jako outlook, thunderbird. Do stejných formátů souborů může být umožněn i export. Statistickou část lze vylepšit o automatizovanou zpětnou vazbu, data miningové metody a prezentaci statistik v podobě grafů. Celý systém by bylo dobré doplnit o vyhledávací mechanismus. Za úvahu stojí i myšlenka migrace Eclipse RCP klienta do webového prostředí. V současnosti existují alespoň dvě možnosti, jak lze migraci provést, použitím projektu Eclipse RAP, či pomocí projektu Eclipse Riana, který vyžaduje hlubší zásah do systému. Po integraci přihlašovacího mechanismu do Administrační konzole by klienti pak mohli sami obsluhovat systém buď pomocí klienta nebo kdekoliv pomocí webové aplikace.

# Literatura

- [1] STERNE J., PRIORE A. Email marketing: using email to reach your target audience and build customer relationships. John Wiley & Sons, 2000. 303 s. ISBN 0471383090.
- [2] JENKINS S. The Truth About Email Marketing. FT Press, 2008. 206 s. ISBN 0789737949.
- [3] MCAFFER J., LEMIEUX J.M. Eclipse Rich Client Platform: Designing, Coding, and Packaging Java™ Applications, Addison Wesley Professional, 2005. 552 s. ISBN 0321334612.
- [4] GALLARDO D., BURNETTE E., MCGOVERN R. Eclipse in Action: A Guide for Web Developers, Manning Publications Co., 2003. 383 s. ISBN 1930110960.
- [5] BAUER Ch., KING G., Java Persistence with Hibernate, Manning Publications Co., 2006. 841 s. ISBN 1932394885.
- [6] GURUMURTHY K. Pro Wicket, Apress, 2006. 304 s. ISBN 1590597222.
- [7] DASHORST M., HILLENIOUS E. Wicket in Action, Manning Publications Co., 2008. 364 s. ISBN 1932394982.
- [8] JavaMail™ API Design Specification Version 1.4, Sun Microsystems, Inc., 2005. 128 s.
- [9] Email Marketing Wiki [online]. last revision 6 June 2008 [cit. 2009-04-20]. Dostupný z WWW: <<http://emailmarketingwiki.org>>.
- [10] Apache Maven Project [online]. 2002-2009 [cit. 2009-04-20]. Dostupný z WWW: <<http://maven.apache.org/>>.
- [11] Reliable Open Source Caucho [online]. 1998-2009 [cit. 2009-04-20]. Dostupný z WWW: <<http://caucho.com/>>.
- [12] RFC 822: Standard for the format of arpa internet text messages [online]. August 13, 1982 [cit. 2009-04-20]. Dostupný z WWW: <<http://www.ietf.org/rfc/rfc0822.txt>>.
- [13] REDMOND, Eric. The Maven 2 POM demystified: The evolution of a project model. *JavaWorld: Solutions for Java Developers* [online]. 29.5.2006 [cit. 2009-04-27]. Dostupný z WWW: <<http://www.javaworld.com/javaworld/jw-05-2006/jw-0529-maven.html>>.